Flow2Seg: Motion-Aided Semantic Segmentation

Xiangtai Li[1], Jiangang Bai[1], Kuiyuan Yang[2], and Yunhai Tong[1]

[1] Key Laboratory of Machine Perception, MOE, School of EECS, Peking University
{lxtpku, pku_bjg, yhtong}@pku.edu.cn
[2] DeepMotion kuiyuanyang@deepmotion.ai

Abstract. Motion is an important clue for segmentation. In this paper, we leverage motion information densely represented by optical flow to assist the semantic segmentation task. Specifically, our framework takes both image and optical flow as input, where image goes through a state-of-the-art deep network and optical flow goes through a relatively shallow network, and results from both paths are fused together in a residual manner. Unlike image, optical flow is weakly related to semantics but can separate different objects according motion consistency, which motivates us to use relatively shallow network to process optical flow to avoid overfitting and keep spatial information. In our experiment on Cityscapes, we find that optical flow improves image-based segmentation on object boundaries especially on small thin objects. Aided by motion, we achieve comparable results with state-of-the-art methods.

Keywords: Optical flow, Flow2Seg, Semantic segmentation

1 Introduction

Semantic segmentation is a fundamental task in computer vision, which aims to predict a semantic category for each pixel in an image. Such comprehensive image understanding is valuable for many vision-based applications such as autonomous driving, remote sensing, human-computer interaction and virtual reality.

In the deep learning era, semantic segmentation has made steady progress after the introduction of Fully Convolutional Networks (FCNs) [24]. However, most existing methods only take a static image as input and ignore the rich motion information in image sequences.

Motion is an important clue for segmentation task and can separate different objects apart based their different motion patterns, which is complementary to static patterns in an image. Motivated by this, we propose to add one path network named Flow2Seg by taking optical flow as input, in addition to the image path modeled by a state-of-the-art network. Fig 1 presents our basic idea. We use flownet [14] to extract optical flow between video frames and use FCN [24] to learn semantic segmentation map directly from optical flow. Considering optical flow is weakly related to semantics and contains lots of noises, we use relatively shallow network to process optical flow to avoid overfitting and

keep spatial information, the design is also empirically verified through ablation study. On the widely used semantic segmentation benchmark Cityscapes [6], Flow2Seg improves the image-based baseline significantly and achieves comparable performance with state-of-the-arts methods. Notably, Flow2Seg improves segmentation of object boundaries, which is crucial for real-world tasks which require to know precise object boundary.

In summary, we propose to use motion information for semantic segmentation task via specifically design network. By combining the new designed motion path with the single frame path modeled by a state-of-the-method segmentation network, we achieve better performance. To the best of our knowledge, we are the first to use network to learn semantic segmentation map directly from optical flow input. Our main contribution can be listed in two points:

1. We propose a novel and light module Flow2Seg for directly mapping optical flow into segmentation map. Combined with the image segmentation model, we achieve considerable improvement compared with the PSP-net [42] baseline on Cityscapes dataset [6]. When training with coarse data, our method achieves 81.4% mIoU which is the top performance compared with other video semantic segmentation methods.

2. We explore the usage of FCNs for learning semantic segmentation map directly from optical flow. Optical flow itself contains little appearance information and we show shallow network can learn better segmentation result than deep models. In addition, we try different optical flow prediction methods, and find that optical flow predicted FlowNet2 [14] contains more detailed information and achieves better results than others.

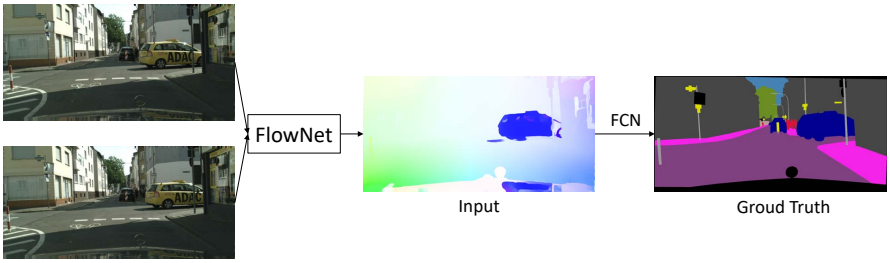

Fig. 1. Overview of the Flow2Seg path. Two consecutive frames are used to estimate the optical flow, then the optical flow is fed into a FCN for semantic segmentation.

2 Related Work

In this section, we briefly review recent works for advancing semantic segmentation from three directions, i.e., context modeling, multi-level feature fusion and using temporal information.

Context Modeing: Contextual information is modeled to gather information from a larger receptive field. ParseNet [23] utilizes global pooling to encode contextual information, and PSPNet [42] uses spatial pyramid pooling to aggregate multi-scale contextual information. Deeplab series [2–4] develop atrous spatial pyramid pooling (ASPP) to capture multi-scale contextual information by dilated convolutional layers with different dilation rates. Instead of parallel aggregation as PSPNet and Deeplab, Yang et al. [36] and Bilinski et al. [1] follow the idea of dense connection [13] to encode contextual information in a dense way. In [27], factorized large filters are directly used to increase the receptive field size for context modeling. In PSANet [43], contextual information is collected from all positions according to the similarities defined in a projected feature space.

Multi-level feature fusion: In addition to contextual information, high-resolution features are also important for high-resolution prediction demanded in semantic segmentation. Accordingly, multi-level feature fusion becomes a common way to use both high-level/low-resolution and low-level/high-resolution features. U-Net [28] adds skip connections between the encoder and decoder to reuse low level features, [41] improves U-Net by fusing high-level features into low-level features. DeepLabV3+ [5] improves the decoder of the previous version by combing low-level features. In [21], Conv-LSTM [34] is proposed to fuse features between layers bidirectionally. Some works fuse different modalities for better performance. PAD-Net [35] is proposed to use gates to fuse multi-modal features trained from multiple auxiliary tasks. Le et at. [19] combines optical flow and surface normals to learn joint multimodal features. Different from their approach, our method uses label map to supervise the optical flow learning process and fuse into the image path in a residual way.

Using temporal information: Sequential frames contain more information than a signal frame, thus temporal modeling is also a promising direction. Fayyaz et al. [9] apply a spatial-temporal LSTM on per-frame CNN features. Nilsson et al.[25] proposed spatio-temporal transformer gated recurrent units (STGRU) to propagate semantic labels bidirectionally towards center frame using optical flow. Jin et al. [16] proposed to learn discriminative features by predicting future frames and combine both the predicted results and current features to parse a frame. Gadde et al. [11] proposed to combine the features wrapped from previous frames with flows and those from the current frames to predict the final results. However, all these methods model temporal motion in an implicit way. For example, Netwarp [11] and STGRU [25] use optical flow to warp features for temporal consistency. One drawback of those method is that they use optical flow to warp feature from previous or future frames and if we suppose the optical flow is accurate, the warped feature is directly matched with current feature and no extra information is added. Our methods focus on learning segmentation maps directly from the raw optical flow inputs and bring motion information as extra guidance.

3 Proposed Method

In this section, we describe our proposed framework in detail. The overall network architecture is shown in Fig 2, which consists of three parts: two independent fully convolutional networks with RGB image and optical flow as input respectively, and one fusion module to learn a joint representation for final segmentation output, the whole network is trained end-to-end with one final loss together with several auxiliary losses.

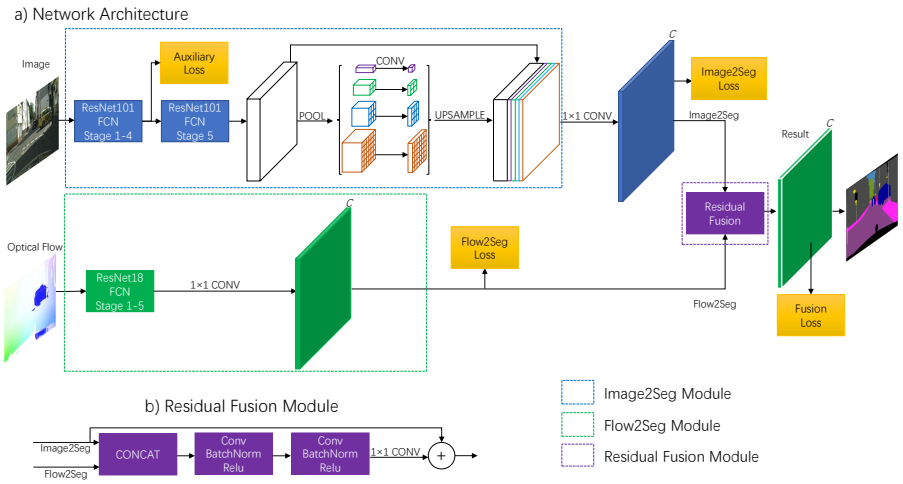

Fig. 2. (a) Network Architecture. It contains three different parts: Image2Seg Module, Flow2Seg Module and Residual Fusion Module. (b) Residual Fusion Module. C denotes the number of categories. Best view it in color.

3.1 Flow2Seg Module

Learning semantic information from optical flow using deep network is first proposed in using a two-stream network [30]. However, unlike video action recognition tasks, pixel-level semantic understanding task needs labeling each pixel rather than only one label for the whole image or optical flow. Thus we adopt fully convolutional network [24] as the feature extractor for the motion stream. In our work, we use off-the-shelf methods for optical flow estimation. In our ablation studies, we found FlowNet2 [14] is more suitable for our task since it can generate sharp object boundaries and generalize well for both small and large motions. Unlike very deep networks those used to extract features for image, relatively shallow ResNet18 is used to process optical flow. To take two channels of optical flow as input for a ImageNet [29] pre-trained ResNet18, we average the

weights of the first convolutional layer along the channel dimension to initialize two-channel input ResNet18. In addition, dilated convolution with dilation rate 2 and 4 are used in stage4 and stage5 to increase the receptive field size while keeping the spatial resolution. The output stride of this network is 8.

Like context modeling for image segmentation, three different context modeling modules are tried after the backbone, including ASPP [4], DenseASPP[36] and pyramid pooling [42], but without observed performance improvement in Flow2Seg. This can be explained that optical flow contains weak semantics and contextual information is not helpful as on image, which is also the reason to choose a relatively shallow network as the backbone. The output of Flow2seg module is a segmentation map with C channels, where C represents the numbers of categories.

3.2 Image2Seg Module

Image2Seg Module maps the input RGB image to semantic segmentation map. Image2Seg module can be any existing FCN architectures [24]. We choose the previous state-of-the-art model PSPNet[42] as our Image2Seg Module. In particular, we use the pretrained ResNet101 [12] with the same dilated strategy as our backbone to extract the feature map. The final feature map size is 1/8 of the input image resolution. On top of the feature map, pyramid pooling module [42] is utilized to incorporate contextual information of multiple levels. Following [42], four average pooling operations with sizes of 1×1, 2×2, 3×3, 6×6 are applied which are represented by different colors in the blue box of Fig 2. Those context features are upsampled to keep the same size with the original feature map by bilinear interpolation, which are further concatenated with the original feature. Then, 1×1 convolution is employed to reduce the feature dimension and fuse the multi-scale context information. Finally, 1×1 convolution is performed on the fused feature map to predict the pixel-level segmentation map. With the same setting as PSPNet [42], auxiliary loss is added after the fourth stage to ease optimization.

3.3 Residual Fusion Module

Both Flow2Seg and Image2Seg generate two semantic maps with C channels in Fig 2 based on two different input modalities, where one is dynamic and the other is static. Though Flow2Seg and Image2Seg are complementary, simply fusing by adding or concatenating their outputs cannot obtain better results since Flow2Seg performs much worse results than Image2Seg. To dig out the useful part in Flow2Seg while discards the useless part, we design a lightweight residual fusion module as illustrated in Fig 2. Both output maps from Flow2Seg and Image2Seg are concatenated together followed by two blocks consisting of convolution and batch normalization [15], and a residual fused semantic map is generated with 1×1 convolution, which is further added to the semantic map generated by Image2Seg for final segmentation. The residual fusion module can refine the weakness part in Image2Seg and leave the well segmented part

unaffected. The output of residual fusion module with fused segmentation maps is the final results of our system.

3.4 Loss Function

As illustrated in Fig 2, the whole network is learned in an end-to-end manner driven by four loss functions defined on four predictions inside the network. In summary, the total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{Image2Seg} + \mathcal{L}_{Fusion} + \alpha * \mathcal{L}_{Flow2Seg} + \beta * \mathcal{L}_{Aux} \tag{1}$$

where $\mathcal{L}_{Image2Seg}$ represents the cross entropy loss between image input results and the ground truth, $\mathcal{L}_{Flow2Seg}$ denotes the cross entropy loss between optical flow input results and the ground truth, $\mathcal{L}_{Aux}$ denotes the auxiliary loss which are used for easy optimization [42, 43] and $\mathcal{L}_{fusion}$ denotes the cross entropy loss between final fusion results and the ground truth, we set $\alpha = 0.2$ and $\beta = 0.4$ respectively in our experiment.

4 EXPERIMENTS

4.1 Datasets

We evaluate the proposed method on Cityscapes [6] which is a standard benchmark for semantic urban scene understanding. It contains 5000 fine pixel-level annotated images, which are divided into 2975, 500, and 1525 images for training, validation and testing, respectively. It also provides 20000 coarsely annotated images. Each finely annotated frame is sampled from the 20th frame of a 30-frame video clip in the dataset, giving in total 180K frames. The previous frame(19th frame) of these images are used for optical flow calculation in our experiment. 30 classes are annotated and 19 of them are used for pixel-level semantic labeling task. Images are high resolution with the same size of 1024×2048. Standard performance metric means Intersection over Union (mIoU) is used for evaluation on both validation set and test set, where labels of test set are not given and predicted results are submitted to server for evaluation.

4.2 Implementation details

Our implementation is based on PyTorch [26], and uses ResNet series as the backbone. In particular, we use ResNet101 as the backbone of Image2Seg and ResNet18 as the backbone of Flow2Seg. We set weight decay to 1e-4, and use Adam [18] as optimizer. We adopt the "poly" learning rate scheduling policy, where initial learning rate is set to 2e-5 and decayed by $(1 - \frac{epoch}{max\_epoch})^{power}$ with $power = 0.9$. Synchronized batch normalization[39]is used for better mean and variance estimation due the limited number of images can be hosted in each GPU. We choose crop size of 832×832 for image input and 1024×1024 for optical flow input. We employ about 100K training iterations with mini-batch size of 8.

Method	mIoU(%)
FlowNetS	35.6
FlowNet2	**39.6**
PWC	36.3
GF-flow	25.4

Table 1. Ablation study with different optical flow inputs, architecture is FCN with ResNet18.

As a common practice to avoid overfitting, data augmentation including random horizontal flipping, random cropping, random color jittering within the range of $[-10, 10]$, and random scaling in the range of $[0.5, 2]$ are used during training and we do these operations for both image and optical flow input. Note for final result submission, we first train the Flow2Seg and Image2Seg independently, then jointly finetune the trained models together with fusion module.

4.3 Experiments on Cityscapes

In this set of experiments except the last experiment, only the 2975 fine annotated images with corresponding optical flows are used for training, and evaluation results on the validation set are reported using single scale prediction. The optical flow is calculated between the current frame and the previous frame. For the last experiment, we also use coarse data to boost our model as well as for fair comparison with other video semantic segmentation methods.

Ablation study on input optical flow We first explore four different methods for optical flow estimation: FlowNetS [10], FlowNet2 [14], PWC [31], and GF-flow [8]. Note that the first three are generated by a trained network. The result is reported in Table 1, FlowNet2 is slightly better than others because it contains more detailed information on object boundaries and more consistent motion on both large and small objects.

Ablation study on architecture of Flow2Seg Module We also explore the different network architectures for Flow2Seg and report the results in Table 2. We first choose different backbone networks from ResNet series, and find that increasing the depth of network decreases the performance. Then we also add context modeling module [42, 4, 36] on the top of ResNet18, and find no performance improvement, which demonstrates that optical flow contains limited semantics and without requiring deep and large contextual modeling.

Comparison with PSPNet baseline We re-implement PSPNet on Cityscapes and achieve similar performance with mIoU of 77.8% which are used as our strong baseline model. We use weights of PSPNet to initialize our Image2Seg module. Then we add our Flow2Seg module together with residual fusion module and

Method	mIoU(%)
ResNet18-FCN	**39.6**
ResNet50-FCN	37.4
ResNet101-FCN	35.4
ResNet18 + ASPP	39.4
ResNet18 + DenseASPP	39.3
ResNet18 + PSP	38.2

Table 2. Ablation study architecture of Flow2Seg Module. Optical flow is generated from FlowNet2. First three rows use different network backbone while last three rows use different context modeling methods.

Method	mIoU(%)
ResNet101-FCN	75.3
ResNet101 + PSP	77.8
ResNet101 + PSP + Flow2Seg	79.7(1.9 ↑)

Table 3. Comparison experiments with baseline on Cityscapes validation set

we train three components together. Finally we get a significant improvement of 1.9% with mIoU of 79.7%, Table 3 summarizes the results. Fig 3 visually compares the segmentation results of PSPNet and our method. We observed that our method improves the object boundaries especially small and thin objects mostly. For example, in the second row of Fig 3, our method can find missing pole in the scene shown in yellow boxes and in the third and fourth rows of Fig 3, our method can handle in-consistent of moving car shown in yellow boxes.

Comparison with state-of-the-art image semantic segmentation methods We first show the comparison between our proposed method and current state-of-the-art image semantic segmentation methods (illustrated in Table 4). Firstly, we train our method only using the train-fine dataset, and achieve better performance than PSPNet [42] and PSANet [43] on the test set. We improve baseline PSPNet [42] by around 1% point. Secondly, we further fine-tune the model with both train-fine and val-fine datasets and get a better performance. Following the same setting as [42], multi-scale sliding-window crop test is used for fair comparison. Detailed per-class results on test set are reported in Table 5. In particular, our method gets superior performance in small objects like "pole","traffic light" and "traffic sign" shown in Table 5 which is consistent with our observation in Fig 3.

Comparison with other video semantic segmentation methods We further compare our method with other video semantic segmentation methods. For fair comparison, we also use coarse data to boost our model accuracy. We start with a trained model on fine dataset and then we use both coarse and fine data to train Image2Seg model for 20 epoch and we fix Flow2Seg path during the

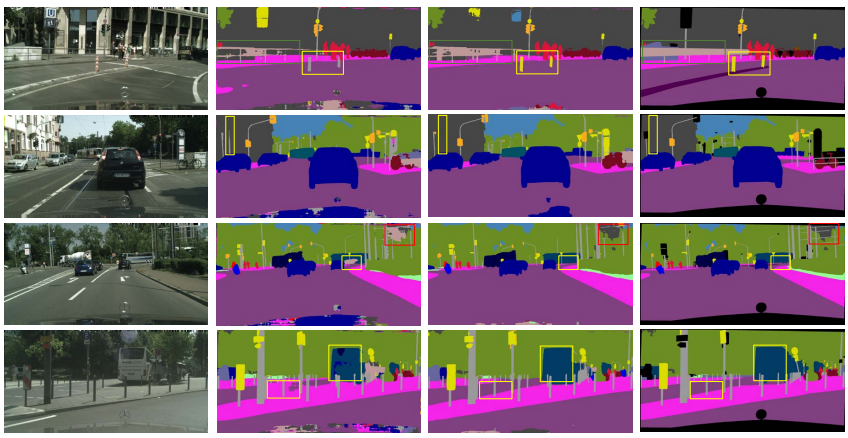

Fig. 3. Comparison of segmentation results of PSPNet and our results on Cityscapes validation set. Our method refines small objects on boarder and generate more consistent results inside objects. Best viewed in color.

Method	Backbone	mIoU(%)
PSPNet [42]†	ResNet101	78.4
PSANet [43]†	ResNet101	78.6
Ours †	ResNet101	**79.4**
RefineNet [22]‡	ResNet101	73.6
SAC [40]‡	ResNet101	78.1
DUC-HDC [32]‡	ResNet101	77.6
AAF [17]‡	ResNet101	79.1
BiSeNet [37]‡	ResNet101	78.9
PSANet [43]‡	ResNet101	80.1
DFN [38]‡	ResNet101	79.3
DSSPN [20]‡	ResNet101	77.8
Ours‡	ResNet101	**80.4**

Table 4. State-of-the-art comparison experiments on Cityscapes test set. †means training with only the train-fine dataset. ‡means training with both the train-fine and val-fine datasets. Note that our methods also use optical flow extracted from the previous frame.

training and then we finetune our model on fine dataset jointly for another 15 epoch. Also, we use multi-scale inference when submitting to the test server. Finally, we achieve 81.4 %mIoU which is the state-of-the art result compared with other video semantic segmentation methods. The results are shown in table 6. Our method performs better than those [11] [16] using flow to warp features which indicates effectiveness of direct motion information.

Method	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
FCN [24]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
DeepLabv2 [3]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
RefineNet [22]	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	69.9	73.6
DSSPN [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	77.8
SAC [40]	**98.6**	86.5	93.1	56.3	59.5	65.1	72.9	78.2	93.5	72.6	95.6	85.9	70.8	95.9	71.2	78.6	66.2	67.7	76.0	78.1
GCN [27]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.9
DUC-HDC [32]	98.5	85.5	92.8	**58.6**	55.5	65.0	73.5	77.8	93.2	72.0	95.2	84.8	68.5	95.4	70.9	78.7	68.7	65.9	73.8	77.6
ResNet38 [33]	98.5	85.7	93.0	55.5	59.1	67.1	74.8	78.7	**93.7**	72.6	**95.5**	86.6	69.2	95.7	64.5	78.8	74.1	**69.0**	76.7	78.4
AAF [17]	98.5	85.6	93.0	53.8	58.9	65.9	75.0	78.4	**93.7**	72.4	95.6	86.4	70.5	95.9	73.9	82.7	76.9	68.7	76.4	79.1
SegModel [7]	**98.6**	**86.4**	92.8	52.4	59.7	59.6	72.5	78.3	93.3	**72.8**	**95.5**	85.4	70.1	95.6	75.4	84.1	75.1	68.7	75.0	78.5
DFN [38]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	79.3
BiSeNet [37]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78.9
PSANet [43]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	80.1
Ours	98.5	85.8	**93.3**	57.6	**63.1**	**68.7**	**76.1**	**80.3**	93.6	72.3	95.4	**87.0**	**72.2**	**96.1**	**75.4**	**88.2**	**77.8**	68.8	**76.4**	**80.4**

Table 5. Per-category results on Cityscapes test set. Note that all the models are trained with only fine-data. Our method outperforms existing approaches in 12 out of 19 categories.

More visible results on video sequence To further prove effectiveness and generality of our method, we show our method results on Cityscapes video clips in Fig 4. We extract optical flow between each frame pair and take both flows and images as inputs. Compared with baseline PSPNet, our method can find missing objects like poles and eliminate ambiguities in the same truck. Since Flow2Seg is a lightweight module with ResNet18 as the feature extractor, our method only costs a little extra computation compared with PSPNet but leads to better performance.

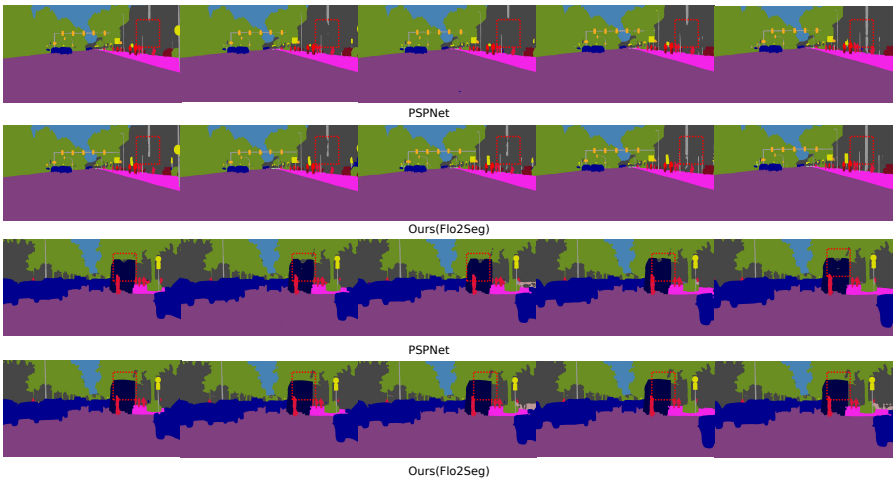

Fig. 4. More comparison of segmentation results of PSPNet and our results on Cityscapes video sequences. The first two rows show our method handles missing small objects on successive frames while the last two rows show our method can remove ambiguities of the same object. Both are shown in red boxes. Best view in color and zoom in.

Method	Backbone	use optical flow	mIoU(%)
Netwarp [11]	ResNet101	yes	80.5
STGRU [25]	ResNet101	yes	80.2
VSPFL [16]	ResNet101	no	79.3
Ours	ResNet101	yes	**81.4**

Table 6. Video semantic segmentation comparison experiments on Cityscapes test set. All the methods use both coarse and fine data.

5 Conclusion

In this paper, Flow2Seg is proposed to use motion information to improve image semantic segmentation. By exploring this module with different optical flows processed by networks with different depths, we achieve comparable results on Cityscapes benchmark. In particular, we find the motion information provided by optical flow can enhance segmentation on object boundaries and small things in the scene. Our method is especially suitable for video semantic segmentation where both successive optical flows and image frames can be used as inputs. We will consider adding multi-frame optical flows into our module as the future work.

References

1. Bilinski, P., Prisacariu, V.: Dense decoder shortcut connections for single-pass semantic segmentation. In: CVPR (2018)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. ICLR (2015)
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. PAMI (2018)
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
7. Falong Shen, Gan Rui, S.Y., Zeng, G.: Semantic segmentation via structured patch prediction, context crf and guidance crf. In: CVPR (2017)
8. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Scandinavian conference on Image analysis. pp. 363–370. Springer (2003)
9. Fayyaz, M., Saffar, M.H., Sabokrou, M., Fathy, M., Klette, R., Huang, F.: Stfcn: spatio-temporal fcn for semantic video segmentation. arXiv preprint arXiv:1608.05971 (2016)
10. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., Van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. arXiv preprint arXiv:1504.06852 (2015)

11. Gadde, R., Jampani, V., Gehler, P.V.: Semantic video cnns through representation warping. In: ICCV (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
13. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
14. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
16. Jin, X., Li, X., Xiao, H., Shen, X., Lin, Z., Yang, J., Chen, Y., Dong, J., Liu, L., Jie, Z., Feng, J., Yan, S.: Video scene parsing with predictive feature learning. In: ICCV (2017)
17. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: ECCV (2018)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Le, H.A., Baslamisli, A.S., Mensink, T., Gevers, T.: Three for one and one for three: Flow, segmentation, and surface normals. arXiv preprint arXiv:1807.07473 (2018)
20. Liang, X., Zhou, H., Xing, E.: Dynamic-structured semantic propagation network. In: CVPR (2018)
21. Lin, D., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Multi-scale context intertwining for semantic segmentation. In: ECCV (2018)
22. Lin, G., Milan, A., Shen, C., Reid, I.D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017)
23. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579 (2015)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
25. Nilsson, D., Sminchisescu, C.: Semantic video segmentation by gated recurrent flow propagation. In: CVPR (2018)
26. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)
27. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters improve semantic segmentation by global convolutional network. In: CVPR (2017)
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. MICCAI (2015)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)
30. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
31. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (2018)
32. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. In: WACV (2018)
33. Wu, Z., Shen, C., van den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. arXiv preprint arXiv:1611.10080 (2016)

34. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NIPS (2015)
35. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: CVPR (2018)
36. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: CVPR (2018)
37. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: ECCV (2018)
38. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: CVPR (2018)
39. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: CVPR (2018)
40. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: ICCV (2017)
41. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: Exfuse: Enhancing feature fusion for semantic segmentation. In: ECCV (2018)
42. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
43. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: ECCV (2018)