# Improving Semantic Segmentation via Decoupled Body and Edge Supervision

Xiangtai Li[1] ⋆, Xia Li[1,2], Li Zhang[3], Guangliang Cheng[4] ⋆⋆, Jianping Shi[4],
Zhouchen Lin[1], Shaohua Tan[1], and Yunhai Tong[1] ⋆⋆

[1] Key Laboratory of Machine Perception, MOE, School of EECS, Peking University
[2] Zhejiang Lab
[3] Department of Engineering Science, University of Oxford
[4] SenseTime Research

**Abstract.** Existing semantic segmentation approaches either aim to improve the object's inner consistency by modeling the global context, or refine objects detail along their boundaries by multi-scale feature fusion. In this paper, a new paradigm for semantic segmentation is proposed. Our insight is that appealing performance of semantic segmentation requires *explicitly* modeling the object *body* and *edge*, which correspond to the high and low frequency of the image. To do so, we first warp the image feature by learning a flow field to make the object part more consistent. The resulting body feature and the residual edge feature are further optimized under decoupled supervision by explicitly sampling different parts (body or edge) pixels. We show that the proposed framework with various baselines or backbone networks leads to better object inner consistency and object boundaries. Extensive experiments on four major road scene semantic segmentation benchmarks including *Cityscapes*, *CamVid*, *KIITI* and *BDD* show that our proposed approach establishes new state of the art while retaining high efficiency in inference. In particular, we achieve 83.7 mIoU % on Cityscape with only fine-annotated data. Code and models are made available to foster any further research (https://github.com/lxtGH/DecoupleSegNets).

**Keywords:** Semantic segmentation, edge supervision, flow field, multi-task learning.

## 1 Introduction

Semantic segmentation is a fundamental task in computer vision that aims to assign an object class label to each pixel in an image. It is a crucial step towards visual scene understanding, and has numerous applications such as autonomous driving [1], image generation [2] and medical diagnosis.

Although the fully convolutional networks (FCNs) [3] have excelled in many major semantic segmentation benchmarks, they still suffer from the following

⋆ Work done while at SenseTime. Email: lxtpku@pku.edu.cn

⋆⋆ Correspond to: chengguangliang@sensetime.com, yhtong@pku.edu.cn

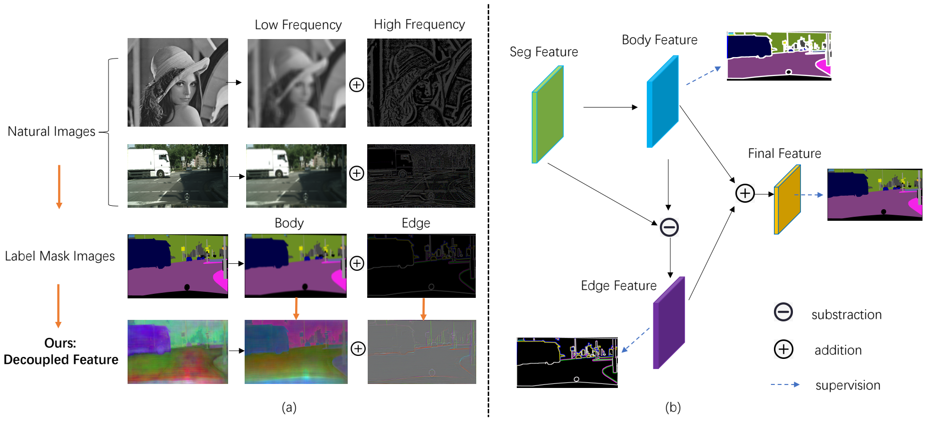

**Fig. 1.** Illustration of our proposed module and supervision framework. (a). The motivation of our proposed framework. Our methods share the same splits with decoupling natural images into low frequency and high frequency. (b).Illustration of our framework. Our method consists of three steps: First, split segmentation feature into body feature and edge feature. Then both parts are supervised with specifically designed loss. Then merge both refined features for the final prediction.

limitations. First, the Receptive Field (RF) of FCNs grows slowly (only linearly) with increasing depth in the network, and such the limited RF is not able to fully model the longer-range relationships between pixels in an image [4, 5]. Thus the pixels are difficult to classify as the ambiguity and noise occurs inside the object body. Moreover, the downsampling operations in the FCNs lead to blurred predictions as the fine details disappear within the significantly lower resolution compared to the original image. As a result, the predicted segments tend to be blobby, and the boundary detail is far from satisfactory, which leads to a dramatic performance drop, especially on small objects.

To tackle the first problem, many approaches [6–8] have been proposed to enlarge the RF, such as dilated convolution, pyramid pooling module [8, 9], non-local operators [10–14], graph convolution network (GCN) [15, 16] and dynamic graph [17]. For the second problem, prior arts manage to embed the low-level features that contain boundary and edge information into high-level features [18–21] or directly refine the outputs [22]. However, the interaction between the object body and object edge is ignored. Can we solve both problems simultaneously?

On the other hand, it is natural for humans to distinguish objects by perceiving both object body and edge information. Inspired by this, we explore the relationships between body and edge in an explicit way to obtain the final semantic segmentation result. As shown in the first two rows of Fig. 1(a), a natural image can be decomposed into a low-spatial frequency component which describes the smoothly changing structure, and a high-spatial frequency component that represents the rapidly changing fine details. This is done by first applying mean or Gaussian filter for smoothing, and the remaining high-frequency parts can be obtained by subtraction. With the same philosophy, a segmentation mask can also be decoupled in this manner, where the finely detailed edge part can be obtained by subtraction from the body part. Inspired by this observation, we assume that a feature map for semantic segmentation can also be decoupled into two parts: body feature and edge feature (see Figure 1(b)). The former contains smooth representation inside the object with low frequency while the latter has sharper details with high frequency.

In this paper, we propose to solve semantic segmentation by explicitly modelling the body consistency and edge preservation in the feature level and then jointly optimizing them in a unified framework. The entire process consists of three steps. First, we propose a novel flow-based method to generate body feature representation by warping each pixel towards object inner parts through a learned offset field to maintain the consistency of body part for each object. Then, we obtain the edge feature by explicitly subtracting the body feature from the input feature. The body feature is supervised by the mask where the edges are ignored during training, while the edge feature is supervised by an edge mask for learning edge prediction. Finally, we merge both optimized features into the final representation for segmentation. As the body generation part is done on a downsampled feature, the edge information is not accurate. We follow the design of [23] to relax the object boundaries during body generation training, which makes both edge and body complementary to each other. Then both parts are merged into a single feature as a reconstructed representation, which is supervised by a commonly used cross-entropy loss. Moreover, the proposed framework is light-weighed and can be plugged into state-of-the-art FCNs [3, 6, 7, 21] based segmentation networks to improve their performance. Our methods achieve top performance on four driving scene semantic segmentation datasets including Cityscapes [24], CamVid [25], KITTI [1] and BDD [26]. In particular, our method achieves 83.7 mIoU on Cityscapes datasets with only fine-annotated data.

The contributions of this paper are as follows,

- We propose a novel framework for the semantic segmentation task by decoupling the body and the edge with different supervisions.
- We propose a lightweight flow-based aggregation module by warping each pixel towards object inner parts through a learned offset field to maintain the consistency of body part for each object.
- Our proposed module can be plugged into state-of-the-art segmentation methods to improve their performance with negligible cost. We carry out extensive experiments on four competitive scene parsing datasets and achieve top performance.

## 2 Related work

**Semantic segmentation.** Recent approaches for semantic segmentation are predominantly based on FCNs [3]. Some earlier works [27–32] use structured prediction operators such as conditional random fields (CRFs) to refine the output boundaries. Instead of these costly DenseCRF, current state-of-the-art methods [7, 8, 10, 12] boost the segmentation performance by designing sophisticated head networks on dilated backbones [6] to capture contextual information. PSPNet [7] proposes pyramid pooling module (PPM) to model multi-scale contexts, whilst DeepLab series [33, 8, 34] uses astrous spatial pyramid pooling (ASPP). In [10, 35, 12, 36, 11, 13], non-local operator [37] and self-attention mechanism [38] are adopted to harvest pixel-wise context from the whole image. Meanwhile, graph convolution networks [39, 15, 40, 17] are used to propagate information over the whole image by projecting features into an interaction space. Different from previous approaches, our method learns a flow field generated by the network itself to warp features towards object inner parts. DCN [41] uses predicted offset to aggregate information in kernel and SPN [42] proposes to propagate information through affinity pairs. Different from both work, our module aims to align pixels towards object inner according to the learned offset field to form body feature which is learned with specific loss supervision. Ding *et al* [43] models unidirectional acyclic graphs to propagate information within the object guided by the boundary. However, it is not efficient due to the usage of the RNN structure between pixels. Our module is light-weighted and can be plugged into the state-of-the-art methods [21, 7] to improve their performance with negligible extra cost, which also proves its efficiency and orthogonality.

**Boundary processing.** Several prior works obtain better boundary localization by structure modeling, such as boundary neural fields [22], affinity field [44], random walk [45]. The work [19, 20] uses edge information to refine network output by predicting edge maps from intermediate CNN layers. However, these approaches have some drawbacks, such as the potential error propagation from wrong edge estimation since both tasks are not orthogonal. Also overfitting edges brings noise and leads to inferior final segmentation results. Zhu et al. [23] proposes boundary relation loss to utilize coarse predicted segmentation labels for data augmentation. Inspired by the idea of label relaxation [23], we supervise the edge and the body parts respectively. The relaxation body avoids the noise from the edge supervision with the relaxation loss. Experimental results demonstrate both higher model accuracy.

**Multi task learning.** Serveral works have proved the effectiveness of combining networks for complementary tasks learning [46, 47]. The works of previous unified architectures that learn a shared representation using multi-task losses. There are some works [18, 48] using learned segmentation and boundary detection network simultaneously and the learned boundaries as an intermediate representation to aid segmentation. GSCNN [18] designs a two-stream network by merging shape information into feature maps explicitly and introduces a dual-task loss to refine both semantic masks and boundary prediction. Different from these works, our goal is to improve the final segmentation results by explicitly optimizing two decoupled feature maps, and we design a specific framework by decoupling semantic body and boundaries into two orthogonal parts with corresponding loss functions and merge them back into final representation for segmentation task.

## 3 Method

In this section, we will first introduce the entire pipeline of our proposed framework in Sec. 3.1. Then we will describe the detailed description of each component in the Sec. 3.2-3.4. Finally, we present the network architectures equipped with our proposed modules and give some discussion on design in Sec. 3.5.

### 3.1 Decoupled segmentation framework

Given a feature map $F \in \mathbb{R}^{H \times W \times C}$, where $C$ represents the channel dimension and $H \times W$ means spatial resolution, our module outputs the refined feature map $\hat{F}$ with the same size. As stated in the introduction part, $F$ can be decoupled into two terms $F_{body}$ and $F_{edge}$. In this paper, we assume they meet the additive rule, which means $F$: $F = F_{body} + F_{edge}$. Our goal is to design components with specific supervision to handle each parts, respectively. We achieve this by first performing body generation and then obtaining the edge part by explicit subtraction where $F_{body} = \alpha(F)$ and $F_{edge} = F - F_{body}$. Then the refined feature $\hat{F}$ can be shown in $\hat{F} = \phi(F) + \varphi(F_{edge}) = F_{body} + \varphi(F - F_{body})$. $\phi$ is the body generation module, which is designed to aggregate context information inside the object and form a clear body for each object. $\varphi$ represents the edge preservation module. We will specify the details of $\phi$ and $\varphi$ in the following sections.

### 3.2 Body generation module

The body generation module is responsible for generating more consistent feature representations for pixels inside the same object. We observe that pixels inside an object are similar to each other, while those lying along the boundary show discrepancy. We propose to explicitly learn body and edge feature representation. To achieve so, we learn a flow field $\delta \in \mathbb{R}^{H \times W \times 2}$, and use it to warp the original feature map to obtain the explicit body feature representation. This module contains two parts: flow field generation and feature warping.

**Flow field generation.** To generate flows that mainly point towards object centers, it is a reasonable way to highlight the features of object center parts as explicit guidance. Generally, low-resolution feature maps (or coarse representation) often contain low-frequency terms. Low spatial frequency parts capture the summation of images, and a lower resolution feature map represents the most salient part where we view it as pseudo-center location or the set of seed points. As shown in Fig. 2(a), we adopt encoder-decoder design where the encoder downsamples feature into low-resolution representation with lower spatial frequency parts. We apply strided-convolutions to compress $F$ into the high-frequency map $F_{low}$. In particular, we adopt three successive $3 \times 3$ depthwise convolution to achieve that. For flow field generation, we share the same pipeline as FlowNet-S [49]. In detail, we first upsample $F_{low}$ to the same size as $F$ via bilinear interpolation, then concatenate them together and apply a $3 \times 3$ convolution layer to predict the flow map $\delta \in \mathbb{R}^{H \times W \times 2}$. Since our model is based

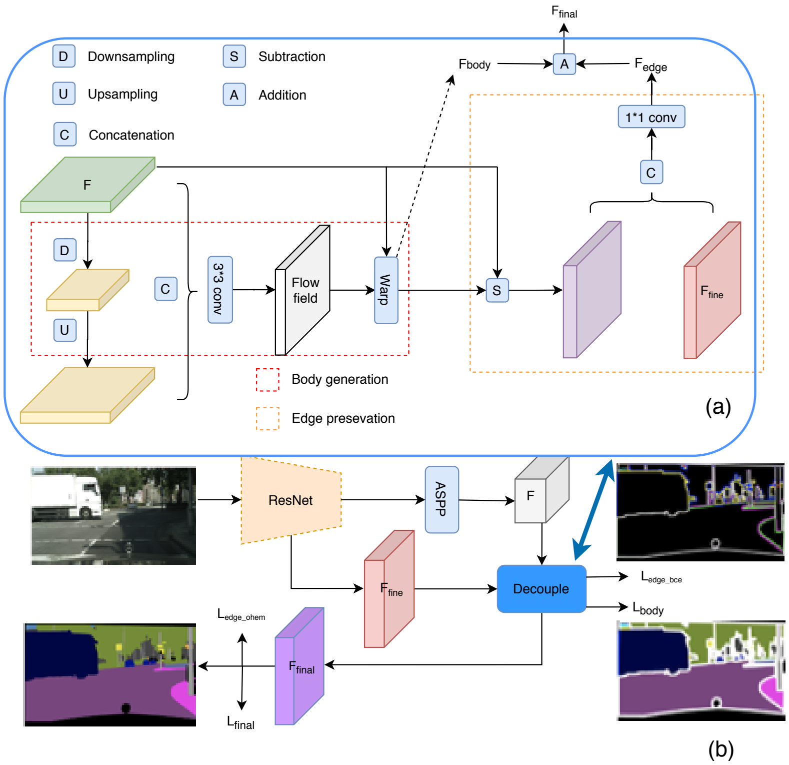

**Fig. 2.** Illustration of our proposed module and supervision framework. (a) shows the proposed decoupled module with Body Generation and Edge Preservation. (b) gives the examples of deploying our methods into Deeplabv3+ [21].

on the dilated backbone network [6], $3 \times 3$ kernel is large enough for covering the long distance between pixels in most cases. More empirical improvements analysis on this implementation can be found in Sec. 4.1.

**Feature warping.** Each position $p_l$ on standard spatial grid $\Omega_l$ is mapped to a new point $\hat{p}$ via $p_l + \delta_l(p_l)$, we then use the differentiable bilinear sampling mechanism to approximate each point $p_x$ in $F_{body}$. The sampling mechanism, proposed in the spatial transformer networks [50, 51], linearly interpolates the values of the four nearest neighbor pixel of $p_l$. The process is shown in Equation 1.

$$F_{body}(p_x) = \sum_{p \in \mathcal{N}(p_l)} w_p F(p) \tag{1}$$

where $w_p$, calculated from flow map $\delta$, represents bilinear kernel weights on warped spatial gird. $\mathcal{N}$ represents the involved neighboring pixels.

### 3.3 Edge preservation module

The edge preservation module is designed to deal with high-frequency terms. It also contains two steps: 1) subtracting the body from the original feature $F$ and 2) adding fine-detailed lower-level features as a supplement. First, we subtract the body feature from the original input $F$. Inspired by recent works on decoder design [21], we add extra low-level feature input as the supplement of missing fine details information to enhance the high-frequency terms in $F_{body}$. Finally, we concatenate both and adopt a $1 \times 1$ convolution layer for fusion. This module can be formulated as Equation 2, where $\gamma$ is a convolution layer and $||$ denotes the concatenation operation.

$$F_{edge} = \gamma((F - F_{body})||F_{fine}) \tag{2}$$

### 3.4 Decoupled body and edge supervision

Instead of supervising the final segmentation map only, we jointly supervise all three parts, including $F_{body}$, $F_{edge}$ and $\hat{F}$, since each part has a specific purpose in our design. In particular, we append auxiliary supervised losses for $F_{body}$ and $F_{edge}$,receptively. For the edge preservation module, we predict a boundary map $b$ during training, which is a binary representation of all the outlines of objects and stuff classes in the images. The total loss $L$ is computed as:

$$L = \lambda_1 L_{body}(s_{body}, \hat{s}) + \lambda_2 L_{edge}(b, s_{final}, \hat{b}, \hat{s}) + \lambda_3 L_{final}(s_{final}, \hat{s}) \tag{3}$$

where $\hat{s}$ represents the ground-truth (GT) semantic labels and $\hat{b}$ is the GT binary masks which is generated by $\hat{s}$. $s_{body}$ and $s_{final}$ denote segmentation map prediction from $F_{body}$ and $F_{final}$ respectively. $\lambda_1$, $\lambda_2$, $\lambda_3$ are three hyper-parameters that control the weighting among the three losses and we set them 1 as default. Note that $L_{final}$ is a common cross entropy loss for segmentation task and we detail the first two items as follows.

To make the optimization easier, for the $F_{body}$ part training, we relax the object boundaries during the training. We use the boundaries relaxation loss [23], which only samples part of pixels within the objects for training.

For the edge part, we propose an integrated loss based on the boundaries edge prior which is got from edge prediction part. For semantic segmentation, most of the hardest pixels to classify lie on the boundary between object classes. Moreover, it is not easy to classify the center pixel of a receptive field when potentially half or more of the input context could be a new class. To tackle this problem, we propose to use such edge prior to handling the boundary pixels particularly and perform online hard example mining at a given edge threshold $t_b$ during the training. The total loss contains two terms: $L_{bce}$ is the binary cross-entropy loss for the boundary pixel classification, while $L_{ce}$ represents cross-entropy loss on edges parts in the scene. The formulations are shown in Equation 4 and Equation 5.

$$L_{edge}(b, s, \hat{b}, \hat{s}) = \lambda_4 L_{bce}(b, \hat{b}) + \lambda_5 L_{ce}(s, \hat{s}, b) \tag{4}$$

$$L_{ce}(s, \hat{s}, b) = -\frac{1}{K} \sum_{i=1}^{N} w_i \cdot \mathbb{1}[s_{i,\hat{s}_i} < t_K \cap \sigma(b_i) > t_b] \cdot \log s_{i,\hat{s_i}} \tag{5}$$

For $L_{edge}$, we set $\lambda_4 = 25$ and $\lambda_5 = 1$ to balance the amount of pixels on edges. For $L_{ce}$, we combine weighted bootstrapped cross-entropy with edge prior from $b$. We set $K = 0.10 \cdot N$, where $N$ is the total number of pixels in the image and $\hat{s}_i$ is the target class label for pixel $i$, $s_{i,j}$ is the predicted posterior probability for pixel $i$ and class $j$, and $\mathbb{1}[x] = 1$ if $x$ is true and 0 otherwise. $\sigma$ represents the Sigmoid function to indicate whether $s$ is on the boundary. The threshold $t_K$ is set in a way that only the pixels with K highest losses are selected while the threshold $t_b$ is to mask non-boundary pixels. Both loss $L_{body}$ and $L_{edge}$ work complementary with each other by sampling pixels separately from different regions in the image. Such design benefits the final performance shown in the experimental parts.

### 3.5 Network architecture

Fig. 2 illustrates the whole network architecture, which is based on state-of-the-art model Deeplabv3+ [21]. Here we utilize dilated ResNet as backbone [52, 6] only for illustration purpose. In particular, our module is inserted after the ASPP module [21]. The decoupled supervisions are appended at the end of decouple module respectively. Moreover, our module is lightweight and can be deployed upon any FCN architectures such as PSPNet [7] to refine the feature representation. When deployed on the native FCN, it is appended after the final output layer of the backbone. When deployed on the PSPNet, it is appended after the PPM module [7]. $F_{fine}$ shares the same design with Deeplabv3+ for both architectures.

## 4 Experiment

**Experiment settings and evaluation metrics:** We first carry out experiments on the Cityscapes dataset, which comprises a large, diverse set of high resolution (2048 × 1024) images recorded in street scenes. It consists of $5,000$ images with high-quality pixel-wise annotations for 19 classes, which is further divided into 2975, 500, and 1525 images for training, validation and testing. To be noted, $20,000$ coarsely labeled images provided by this dataset are **not** used. Furthermore, we also evaluate our methods on CamVid [25], KITTI [1] and BDD [26] datasets. For all datasets, we use the standard mean Intersection over Union (mIoU) metric to report segmentation accuracy. For Cityscapes, we also report F-score proposed in [53] by calculating along the boundary of the predicted mask given a small slack in the distance to show the high-quality segmentation boundaries of predicted mask.

| Method | Backbone | mIoU (%) | $\Delta$(%) |
|---|---|---|---|
| FCN naive | ResNet-50 | 75.4 | - |
| + US [23](Baseline) | ResNet-50 | 76.6 | - |
| + ours | ResNet-50 | **80.1** | **3.5**↑ |
| + US [23](Baseline) | ResNet-101 | 77.8 | - |
| + ours | ResNet-101 | **80.8** | **3.0**↑ |

(a) Ablation study on strong FCN baselines.

| Method | $L_{body}$ | $L_{bce}$ | $L_{edge-ohem}$ | mIoU (%) | $\Delta$(%) |
|---|---|---|---|---|---|
| FCN | | | | 76.6 | - |
| +(BG & EP) | - | - | - | 78.3 | 1.7 ↑ |
| | ✓ | - | - | 78.8 | 0.5↑ |
| | - | ✓ | - | 78.3 | - |
| | - | ✓ | ✓ | 78.7 | 0.4↑ |
| | ✓ | ✓ | ✓ | **80.1** | **1.8**↑ |
| w/o $F_{fine}$ | ✓ | ✓ | ✓ | 79.3 | 0.8 ↓ |
| w/o *ohem* | ✓ | ✓ | × | 79.0 | 1.1 ↓ |

(b) Ablation study on Decoupled Supervision.

| Method | mIoU (%) | $\Delta$(%) |
|---|---|---|
| FCN + BG & EP (Baseline) | 78.3 | - |
| w/o BG warp | 76.9 | 1.4 ↓ |
| w/o BG encoder-decoder | 77.3 | 1.0 ↓ |
| w/o EP | 77.9 | 0.4 ↓ |

(c) Ablation Study on effect of each component.

| Method | mIoU (%) | $\Delta$(%) |
|---|---|---|
| FCN (Baseline) | 76.6 | - |
| w SPN [42] | 77.9 | 1.3↑ |
| w DCN [41] | 78.2 | 1.6 ↑ |
| +GSCNN [18] | 77.8 | 1.2↑ |
| ours | **80.1** | **3.5**↑ |

(d) Comparison to related methods.

**Table 1.** Experimental results on the Cityscapes validation set with dilated FCN as baselines.

**Implementation details:** We use PyTorch [54] framework to carry out the following experiments. All networks are trained with the same setting, where stochastic gradient ‘xsdescent (SGD) with a batch size of 8 is used as the optimizer, with the momentum of 0.9, the weight decay of $5e-4$ and the initial learning rate of 0.01. As a common practice, the ‘poly’ learning rate policy is adopted to decay the initial learning rate by multiplying $(1-\frac{\text{iter}}{\text{total\_iter}})^{0.9}$ during training. Data augmentation contains random horizontal flip, random resizing with scales range of $[0.75,\ 2.0]$, and random cropping with size 832. Specifically, we use ResNet-50, ResNet-101 [52] and Wider-ResNet [55] as the backbones. Additionally, we re-implement the state-of-the-arts [7, 21] for fairness. We run the training for 180 epochs for ablation purposes, and we run 300 epochs for the submission to the test server. We first train the base networks without our module as initial weights and then train with our framework with the same epoch for each experiment. All the models are evaluated with the sliding-window manner for a fair comparison.

### 4.1 Ablation studies

**Improvements over baseline model.** We first apply our method on naive dilated FCN models [6], where we also include uniform sampling trick [23] to balance classes during training as our strong baselines in Table 1(a). Our naive FCNs achieve 76.6 and 77.8 in mIoU for ResNet-50 and ResNet-101, respectively. After applying our method, we achieve significant improvements over each backbone by 3.5% and 3.0%, respectively. Note that our ResNet-50 based model is 2.2% higher than ResNet-101 baseline, which indicates the performance gain is not from the more convolution layers in Body Generation.

**Ablation studies on decoupled supervisions.** Then we explore the effectiveness of decoupled supervision in Table 1(b). Directly adding both the body generation and edge preservation module results in a 1.7% improvement, which

| Method | Backbone | mIoU (%) | $\Delta$(%) | FLOPS |
|---|---|---|---|---|
| PSPNet [7] | ResNet-50 | 79.6 | - | 132.1 |
| +ours | ResNet-50 | 81.0 | 1.4↑ | + 9.2 (6.8%) |
| Deeplabv3+ [21] | ResNet-50 | 79.7 | - | 190.1 |
| +ours | ResNet-50 | 81.5 | 1.8↑ | +9.0 (4.7%) |
| Deeplabv3+ [12] | Wider-ResNet | 81.3 | - | 664.5 |
| +ours | Wider-ResNet | 82.4 | 1.1↑ | +7.5 (1.1%) |

(a) Improvements upon different state-of-the-arts. To compute FLOPS, we adopt 512 × 512 images as the input.

| Method | mIoU(%) |
|---|---|
| Deeplabv3+ (ResNet-50) | 79.7 |
| + BG & EP (ResNet-50) | 81.5 (1.8↑) |
| Deeplabv3+ (ResNet-101) | 80.7 |
| + BG & EP (ReseNet-101) | 82.6 (1.9↑) |
| + BG & EP (ReseNet-101) +MS | 83.5 |

(b) Ablation study on improvement strategy on the validation set.

**Table 2.** Experiment results on Cityscapes validation set with more network architectures. Best viewed in color and zoom in.

| Method | Backbone | mIoU (%) |
|---|---|---|
| AAF [44] | ResNet-101 | 79.1 |
| PSANet [56] | ResNet-101 | 80.1 |
| DFN [57] | ResNet-101 | 79.3 |
| DenseASPP [34] | DenseNet-161 | 80.6 |
| DAnet [10] | ResNet-101 | 81.5 |
| CCNet [11] | ResNet-101 | 81.4 |
| BAFNet [43] | ResNet-101 | 81.4 |
| ACFNet [58] | ResNet-101 | 81.9 |
| GFFnet [59] | ResNet-101 | 82.3 |
| Ours | ResNet-101 | **82.8** |

(a) Results on Cityscapes test server trained with only fine-data.

| Method | Coarse | Backbone | mIoU (%) |
|---|---|---|---|
| PSP [7] | ✓ | ResNet-101 | 81.2 |
| Deeplabv3+ [21] | ✓ | Xception | 82.1 |
| DPC [60] | ✓ | Xception | 82.6 |
| Auto-Deeplab [61] | ✓ | - | 82.1 |
| Inplace-ABN [62] | ✓ | Wider-ResNet | 82.0 |
| Video Propagation [23] | ✓ | Wider-ResNet | 83.5 |
| G-SCNN [18] | × | Wider-ResNet | 82.8 |
| Ours | × | Wider-ResNet | **83.7** |

(b) Results on the Cityscapes test server.

**Table 3.** Comparison with state-of-the-art on the Cityscapes test set. To be noted, our method **does not use coarse data.**

shows its aligned effect. After appending $L_{body}$, we get an obvious improvement of 0.5%, and it can avoid uncertain noises on boundaries. $L_{bce}$ has no effect on the final performance since there is no direct supervision to segmentation prediction. Adding $L_{ce}$ will bring about 0.4% improvement, which indicates that our integrated loss can better mine the boundaries based shape prior. Finally, after combining all three losses, we get a higher improvement by 1.8%, which demonstrates the orthogonality of our separated supervision design. We also remove the $F_{fine}$ module in Equation 2, which results in about 0.8% drop in the final performance. This indicates the effectiveness of the edge cue from low-level features. Meanwhile, we also remove the hard pixel mining on $L_{edge-ohem}$, which results in about a 1.1% drop. That shows the effectiveness of our proposed integrated loss on boundaries.

**Ablation study on the effect of each component.** Here we carry out more detailed explorations on our component design with no the decoupled supervision setting shown in Table 1. Removing warping in BG achieves 76.9 in mIoU, which is a big decrease while removing the encoder-decoder part of BG results in 77.3% due to the limited receptive field of dilated FCN. Removing EP leads to less performance drop as the main drop of FCN is on large objects shown in table 2.

**Comparison with related methods.** To verify the effectiveness of the BG module, we replace our BG module with DCN and SPN operators. The former is used to aggregate features with learned offset field, while the latter propagate information through learned affinity pair. The first two rows in Table 1(f) demonstrates that our method works better than DCN and SPN, which proves

| Method | mIoU | road | swalk | build. | wall | fence | pole | tlight | sign | veg | terrain | sky | person | rider | car | truck | bus | train | motor | bike |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FCN [6] | 76.6 | 98.0 | 84.5 | 92.5 | 50.7 | 62.7 | **67.7** | 73.8 | 81.2 | **92.8** | 61.2 | **94.7** | **83.8** | 64.2 | 95.0 | 56.4 | 81.6 | 60.5 | 68.2 | 79.4 |
| Ours | **80.1** | **98.4** | **86.4** | **92.9** | **58.7** | **64.8** | 67.0 | **74.3** | **82.2** | 92.7 | **63.0** | 94.5 | 83.6 | **66.2** | **95.2** | **78.6** | **91.0** | **83.2** | **69.3** | **79.7** |
| PSPNet [7] | 79.6 | 98.0 | 84.5 | 92.9 | 54.9 | 61.9 | 66.5 | 72.2 | 80.9 | 92.6 | **65.6** | **94.8** | 83.1 | 63.5 | 95.4 | 83.9 | 90.6 | 84.01 | 67.6 | 78.5 |
| Ours | **81.0** | **98.2** | **85.8** | **93.4** | **59.5** | **67.0** | **68.7** | **74.6** | **81.6** | **92.8** | 65.5 | 94.3 | **83.6** | **65.6** | **95.6** | **86.7** | **92.6** | **87.1** | **68.6** | **79.1** |
| Deeplabv3+ [21] | 79.7 | 98.2 | 85.3 | 92.8 | 58.4 | 65.4 | 65.6 | 70.4 | 79.2 | 92.6 | 65.2 | 94.8 | 82.4 | 63.3 | 95.3 | 83.2 | 90.7 | 84.1 | 66.1 | 77.9 |
| Ours | **81.5** | **98.3** | **86.5** | **93.6** | **60.7** | **66.8** | **70.7** | **73.9** | **81.9** | **93.1** | **66.1** | **95.2** | **84.3** | **67.5** | **95.8** | **86.1** | **92.3** | **85.5** | **72.1** | **80.1** |

**Table 4.** Per-category results on the Cityscapes validation set. Note that our method improves all strong baselines in most categories.

the effectiveness of our BG module design. We also compare with GSCNN [18] in the same setting.

**Improvements upon different base models.** To further verify the generality of our proposed framework, we test it upon several state-of-the-art models including PSPNet [7] and Deelabv3+ [21] with various backbone network in Table 2(a). It can be seen that our method improves those by around 0.9%-1.5% in mIoU. Note that our baselines are stronger than the original paper. Meanwhile, we also report the FLOPS during the inference stages in the last column of Table 2(a). Our module is extremely lightweight with only 1.1%-6.8% relative FLOPS increment.

**Comparison to state-of-the-arts.** For fair comparison, we follow the common procedure of [10, 11, 18] including stronger backbone (ResNet-101 [52]) and multi-scale inference (MS) to improve the model performance. As shown in Table 2(b), our best model achieves 83.5 mIoU on the validation dataset after applying both techniques. Then we compare our method with state-of-the-arts on the Cityscapes test set in Table 3 using the best model in Table 1. We first report results using ResNet-101 backbone in Table 3(a) and our method achieves **82.8** mIoU which improves by a large margin over all previous works. Moreover, we further apply our methods with a stronger backbone Wider-ResNet [55] pretrained on the Mapillary [63] dataset, which shares the same setting with GSCNN [18]. Our method achieves **83.7** in mIoU and also leads to a significant margin over GSCNN [18]. Table 3(b) shows the previous state-of-arts, which also uses much large coarse video data [23]), while our method achieves much better performance with utilizing only fine-annotated data.

### 4.2 Visual analysis

**Improvement analysis.** Here we illustrate a detailed analysis of improvements. First we report mIoU of each category in Table 4. For the FCN model, our method improves a lot on large objects like bus and car in the scene. For Deeplabv3+ and PSPNet models, our method improves mainly on small objects such as traffic light and pole since most large patterns are handled by context aggregation modules like PPM and ASPP. To be more specific, we also evaluate the performance of predicted mask boundaries shown in Fig. 3, where we report the mean F-score of 19 classes at 4-different thresholds. From that figure, we conclude that our methods improve the baseline object boundaries by a significant margin and our method is also slightly better than GSCNN [18] on

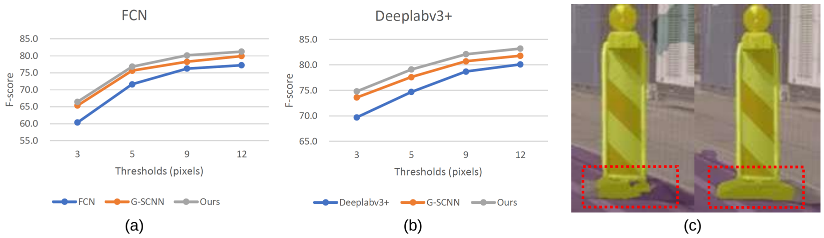

**Fig. 3.** Improvement analysis on boundaries where F-score is adopted. (a) is the improvement on FCN. (b) is the improvement on Deeplabv3+. (c) is the improvement on mask boundary prediction. Best viewed in color and zoom in.

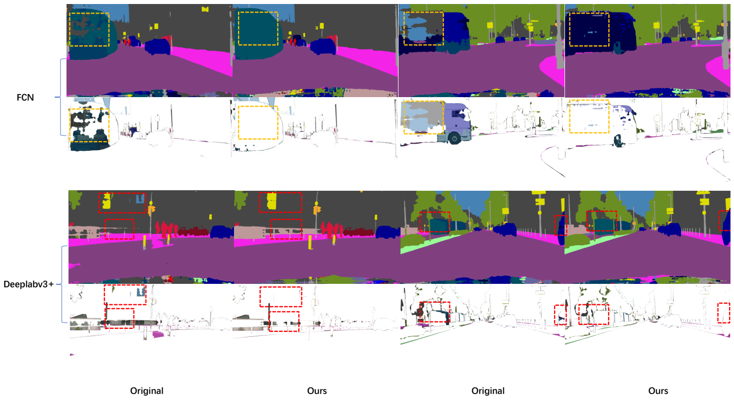

**Fig. 4.** Comparisons upon on FCN and Deeplabv3+. The first and third rows are prediction masks while the second and last rows are error maps compared with the ground truth mask. The first two rows illustrate the results on FCN and ours, while the last two rows show Deeplabv3+'s and ours'. Our method solves the inner blur problem in large patterns for FCN shown in yellow boxes and fixes missing details and inconsistent results on small objects on Deeplabv3+ shown in red boxes.

both different cases with four different thresholds. To be noted, we compared deeplabv3+ ResNet101 backbone in Fig. 3(b) with original paper while GSCNN results in Fig. 3(a) with the ResNet50 backbone implemented by us. Fig. 3 (c) shows some visual examples of our model prediction with a more precise boundary mask. Fig. 4 presents three visual examples over error maps. Our methods can better handle the inconsistency on large objects in FCN and boundaries of small objects in Deeplabv3+, which follows the same observation in Table 4. More visual examples can be found in the supplementary file.

**Visualization on decoupled feature representation and prediction.** We visualize the decoupled feature and prediction masks of our model over Deeplabv3+ in Fig. 4. Figures in (a)-(c) are drawn by doing Principal Component Analysis (PCA) from feature space into RGB space. As shown in Fig. 5, the features in

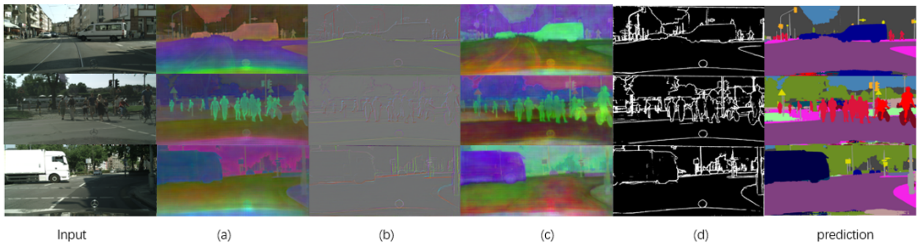

**Fig. 5.** Visualization results based on Deeplabv3+ models. (a) is $F_{body}$. (b) is $F - F_{body}$. (c) is re-constructed feature $\hat{F}$. (d) is edge prior prediction $b$ with $t_b = 0.8$. Best viewed in color and zoom in.

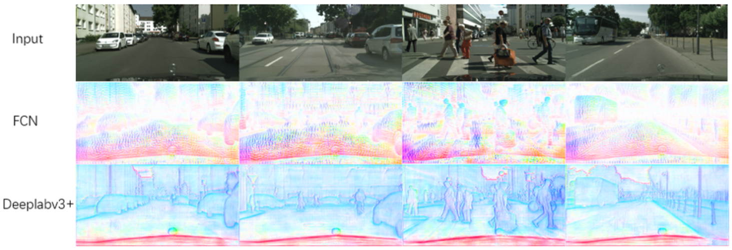

**Fig. 6.** Flow maps visualization. The second row shows learned flow maps based on FCN, while the last row shows the learned maps based on Deeplabv3+.

(a) and (b) are complementary to each other, where each pixel in body part shares similar feature representations while pixels on edge varies. The merged feature in (c) have more precise and enhanced boundaries, while the objects in (a) are thinner than (c) due to boundary relaxation. The predicted edge prior in (d) has a more precise location of each object's boundaries. This gives better prior for mining hardest pixels along the boundaries parts. More visualization examples can be found in the supplementary file.

**Visualization on flow field in BG.** We also visualize the learned flow field for FCNs and Deeplabv3+ in Fig. 6. Both cases differ significantly. For the FCN model, we find that the learned flow field point towards the inner part in each object, which is consistent with our goal stated in Sec. 3.2. While for the Deeplabv3+ model, the learned flow is sparse and mainly lies on the object boundaries because enough context has been considered in the ASPP module. This observation is consistent with the results in Table 4: predictions over large objects are mainly improved in FCN (truck, 22%), while those over small objects are mainly improved in Deeplabv3+ (pole, 5%).

### 4.3 Results on other datasets

To further prove the generality of our proposed framework, we also perform more experiments on the other three road sense datasets. Our model is the same

| Method | Backbone | Pretrain | mIoU(%) |
|---|---|---|---|
| DenseDecoder [64] | ResNext-101 | ImageNet | 70.9 |
| BFP [43] | ResNet-101 | ImageNet | 74.1 |
| Ours | ResNet-101 | ImageNet | **76.5** |
| VideoGCRF [65] | ResNet-101 | Cityscapes | 75.2 |
| Video Propagation [23] | Wider-ResNet | Cityscapes | 79.8 |
| Ours | ResNet-101 | Cityscapes | **81.5** |
| Ours | Wider-ResNet | Cityscapes | **82.4** |

(a) Results on CamVid.

| Method | Backbone | mIoU(%) |
|---|---|---|
| Dilated [26] | ResNet101 | 55.2 |
| FasterSeg [66] | - | 55.3 |
| Ours | ResNet101 | **66.9** |

(b) Results on BDD.

| Method | IoU class(%) | iIoU class(%) | IoU category(%) | iIoU category(%) |
|---|---|---|---|---|
| AHiSS [67] | 61.2 | 26.9 | 81.5 | 53.4 |
| LDN [68] | 63.5 | 28.3 | 85.3 | 59.1 |
| MapillaryAI [62] | 69.6 | 43.2 | 86.5 | 68.9 |
| Video Propagation [23] | **72.8** | 48.7 | **88.9** | 75.2 |
| Ours | **72.8** | **49.5** | 88.5 | **75.5** |

(c) Results on KITTI.

**Table 5.** Experiments results on other road scene benchmarks.

as used in Cityscapes datasets, which is based on Deeplabv3+ [21]. Standard settings of each benchmark are used, which are summarized in the supplementary file for the detailed reference.

**CamVid:** CamVid [25] is another road scene dataset. This dataset involves 367 training images, 101 validation images, and 233 testing images with a resolution of 720 × 960. For a fair comparison, we compare both ImageNet pretrained and Cityscapes pretrained models. As shown in Table 5(a), our methods achieve significant gains over other state-of-the-arts in both cases.

**BDD:** BDD [26] is a new road scene benchmark consisting $7,000$ images for training and $1,000$ images for validation. Compared with baseline model [26] which uses dilated backbone (55.2%), our method leads to about **12**% mIoU improvement with single scale inference with the same ResNet-101 backbone and achieves top performance with 66.9% in mIoU.

**KITTI:** KITTI benchmark [1] has the same data format as Cityscapes, but with a different resolution of $375 \times 1242$ and more metrics. The dataset consists of 200 training and 200 test images. Since it is a small dataset, we follow the settings from previous work [23, 62] by finetuning our best model from the Cityscapes dataset. Our methods rank first on three out of four metrics. It should be noted that method [23] uses both video and coarse data during Cityscapes pretraining process, while we only use the fine-annotated data.

## 5 Conclusions

In this paper, we propose a novel framework to improve the semantic segmentation results by decoupling features into the body and the edge parts to handle inner object consistency and fine-grained boundaries jointly. We propose the body generation module by warping feature towards objects' inner parts then the edge can be obtained by subtraction. Furthermore, we design decoupled loss by sampling pixels from different parts to supervise both modules' training. Both modules are light-weighted and can be deployed into the FCN architecture for end-to-end training. We achieve state-of-the-art results on four road scene parsing datasets, including Cityscapes, CamVid, KITTI and BDD. The superior performance demonstrates the effectiveness of our proposed framework.

## References

1. Andreas, G., Philip, L., Raquel, U.: Are we ready for autonomous driving? the kitti vision benchmark suite. CVPR (2012)
2. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR. (2018)
3. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
4. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint (2014)
5. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: NeurIPS. (2016)
6. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. ICLR (2016)
7. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)
8. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint (2017)
9. Hou, Q., Zhang, L., Cheng, M.M., Feng, J.: Strip pooling: Rethinking spatial pooling for scene parsing. In: CVPR. (2020)
10. Fu, J., Liu, J., Tian, H., Fang, Z., Lu, H.: Dual attention network for scene segmentation. arXiv preprint (2018)
11. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: ICCV. (2019)
12. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: ICCV. (2019)
13. He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: ICCV. (2019)
14. Li, X., Zhang, L., You, A., Yang, M., Yang, K., Tong, Y.: Global aggregation then local distribution in fully convolutional networks. In: BMVC. (2019)
15. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: NeurIPS. (2018)
16. Zhang, L., Li, X., Arnab, A., Yang, K., Tong, Y., Torr, P.H.: Dual graph convolutional network for semantic segmentation. In: BMVC. (2019)
17. Zhang, L., Xu, D., Arnab, A., Torr, P.H.: Dynamic graph message passing networks. In: CVPR. (2020)
18. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. ICCV (2019)
19. Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: CVPR. (2016)
20. Gong, K., Liang, X., Li, Y., Chen, Y., Yang, M., Lin, L.: Instance-level human parsing via part grouping network. In: ECCV. (2018)
21. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. (2018)
22. Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: CVPR. (2016)
23. Zhu, Y., Sapra, K., Reda, F.A., Shih, K.J., Newsam, S., Tao, A., Catanzaro, B.: Improving semantic segmentation via video propagation and label relaxation. In: CVPR. (2019)

24. 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)
25. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters (2008)
26. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR. (2020)
27. 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)
28. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: CVPR. (2016)
29. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.S.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)
30. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: ICCV. (2015)
31. He, X., Gould, S.: An exemplar-based crf for multi-instance object segmentation. In: CVPR. (2014)
32. Jampani, V., Kiefel, M., Gehler, P.V.: Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In: CVPR. (2016)
33. 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)
34. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: CVPR. (2018)
35. He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: CVPR. (2019)
36. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: ICCV. (2019)
37. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR. (2018)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. (2017)
39. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. (2017)
40. Li, X., Yang, Y., Zhao, Q., Shen, T., Lin, Z., Liu, H.: Spatial pyramid based graph reasoning for semantic segmentation. In: CVPR. (2020)
41. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. (2017)
42. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: NeurIPS. (2017)
43. Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. In: ICCV. (2019)
44. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: ECCV. (2018)
45. Bertasius, G., Torresani, L., Yu, S.X., Shi, J.: Convolutional random walk networks for semantic image segmentation. In: CVPR. (2017)
46. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: CVPR. (2016)

47. Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: CVPR. (2017)
48. 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)
49. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: CVPR. (2015)
50. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NeurIPS. (2015)
51. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR. (2017)
52. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
53. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. (2016)
54. 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: NeurIPS Workshop. (2017)
55. Zagoruyko, S., Komodakis, N.: Wide residual networks. (2016)
56. 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)
57. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: CVPR. (2018)
58. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: Acfnet: Attentional class feature network for semantic segmentation. In: ICCV. (2019)
59. Li, X., Houlong, Z., Lei, H., Yunhai, T., Kuiyuan, Y.: Gff: Gated fully fusion for semantic segmentation. In: AAAI. (2020)
60. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., eds.: NeurIPS. (2018)
61. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. CVPR (2019)
62. Rota Bulò, S., Porzi, L., Kontschieder, P.: In-place activated batchnorm for memory-optimized training of dnns. In: CVPR. (2018)
63. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV. (2017)
64. Bilinski, P., Prisacariu, V.: Dense decoder shortcut connections for single-pass semantic segmentation. In: CVPR. (2018)
65. Chandra, S., Couprie, C., Kokkinos, I.: Deep spatio-temporal random fields for efficient video segmentation. In: CVPR. (2018)
66. Chen, W., Gong, X., Liu, X., Zhang, Q., Li, Y., Wang, Z.: Fasterseg: Searching for faster real-time semantic segmentation. In: ICLR. (2020)
67. Meletis, P., Dubbelman, G.: Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation. In: IVS. (2018)
68. Krapac, J., Kreso, I., Segvic, S.: Ladder-style densenets for semantic segmentation of large natural images. In: ICCV Workshop. (2017)