# Joint Semantic Segmentation and Boundary Detection using Iterative Pyramid Contexts

Mingmin Zhen[1] Jinglu Wang[2] Lei Zhou[1] Shiwei Li[3]
Tianwei Shen[1] Jiaxiang Shang[1] Tian Fang[3] Long Quan[1]
[1]Hong Kong University of Science and Technology [2]Microsoft Research Asia [3]Everest Innovation Technology
{mzhen, lzhouai, tshenaa, jshang, quan}@cse.ust.hk
Jinglu.Wang@microsoft.com {sli, fangtian}@altizure.com

## Abstract

*In this paper, we present a joint multi-task learning framework for semantic segmentation and boundary detection. The critical component in the framework is the iterative pyramid context module (PCM), which couples two tasks and stores the shared latent semantics to interact between the two tasks. For semantic boundary detection, we propose the novel spatial gradient fusion to suppress non-semantic edges. As semantic boundary detection is the dual task of semantic segmentation, we introduce a loss function with boundary consistency constraint to improve the boundary pixel accuracy for semantic segmentation. Our extensive experiments demonstrate superior performance over state-of-the-art works, not only in semantic segmentation but also in semantic boundary detection. In particular, a mean IoU score of* $81.8\%$ *on Cityscapes test set is achieved without using coarse data or any external data for semantic segmentation. For semantic boundary detection, we improve over previous state-of-the-art works by* $9.9\%$ *in terms of AP and* $6.8\%$ *in terms of MF(ODS).*

## 1. Introduction

Semantic segmentation has been actively studied in many recent papers and is also critical for various challenging applications such as autonomous driving [1] and virtual reality [2]. In semantic segmentation tasks, we estimate a mask where each pixel represents a category ID (Figure 1). The semantic boundary detection task is a multi-label classification task and different from traditional binary edge detection. As a dual problem of semantic segmentation, which means that the boundary always surrounds the mask, the goal of semantic boundary detection [3, 4] is to identify image pixels that belong to object (class) boundaries. In general, estimating the semantic label at image boundaries is challenging as it could be ambiguous between two sides. The boundary accuracy of this mask is crucial to the final semantic segmentation accuracy, yet its importance is often overlooked in previous methods.

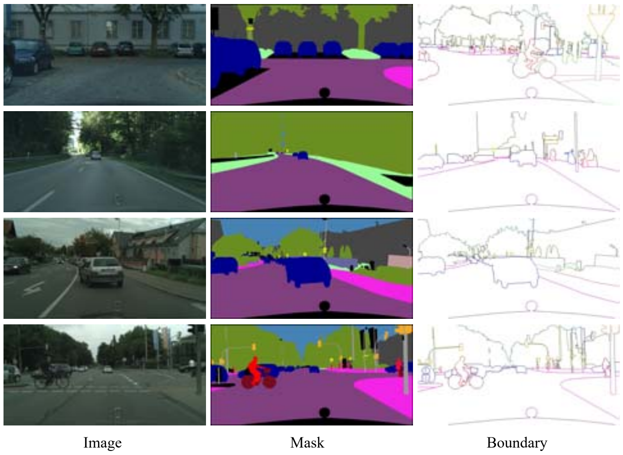

Figure 1. Some samples for semantic segmentation in Cityscape dataset (**best viewed in color**). We follow [3, 4] to generate the semantic boundary.

Recently, some works adopt edge detetion as auxiliary information to improve the performance of semantic segmentation. In [5], a two-stream CNN architecture for semantic segmentation is proposed that explicitly wires shape information as a separate processing branch, i.e., shape stream, that processes information in parallel to the classical stream. [6] learns the edges as an additional semantic class to enable the network to be aware of the boundary layout. Unidirectional acyclic graphs (UAGs) are used to model the function of undirected cyclic graphs (UCGs) in order to overcome the loopy property and improve efficiency. Based on the proposed UAGs, holistic context is aggregated via harvesting and propagating the local features throughout the whole image efficiently. However, these works ignore the semantic information of edges in semantic segmentation. In essence, semantic boundary detection is more highly coupled with semantic segmentation compared with binary edge detection.

In this paper, we take a step further to propose a joint-task framework combining both semantic segmentation and semantic boundary detection. Semantic segmentation task and semantic boundary task are correlated together iteratively, which conforms to the dual relationship between them. We leverage pyramid context information of one task to refine another task.

For semantic boundary detection, one challenging issue is to suppress the non-semantic edges, which are ambiguous to distinguish from semantic edges. To address this problem, we derive the spatial gradient, which is similar to the image gradient method [7], as the initial semantic boundary from the semantic mask and fuse it with the semantic boundary probability map to obtain clean semantic boundary results.

As there exists the duality constraint between semantic segmentation and semantic boundary detection, we propose a novel loss function to enforce boundary consistency for semantic segmentation task. For the predicted mask, boundary is derived as the outer contour, which can be used to constrain the mask. The differences between the prediction results and the groundtruth are then formulated as one loss term, which we refer to as duality loss, to impose boundary consistency on semantic mask during model training. The duality loss term is differentiable due to the pixel-wise operation. The overall network can be trained in an end-to-end manner.

Our experiments show that our proposed method outperforms existing state-of-the-art works, especially for semantic boundary detection task. The proposed method surpasses state-of-the-art work [4] by 9.9% in terms of average precision (AP) and 6.8% in terms of maximum F-measure (MF) at optimal dataset scale (ODS) for semantic boundary detection. We achieve mean intersection over union (mIoU) score 81.8% on Cityscape test set with only fine annotated trainval data used for training. The performance on the validation set is also better than previous works. The significant gains of performance verify the effect of proposed method. All in all, our main contributions can be summarized as follows:

- To our best knowledge, we are the **first** to combine semantic boundary detection task and semantic segmentation task into a joint multiple-task learning framework with iterative pyramid context module (PCM).
- In the semantic boundary detection module, we introduce a novel strategy to suppress non-semantic edges, by fusing the derived boundary from mask probability map with semantic boundary probability map.
- In the semantic segmentation module, We design a duality loss, which improves the boundary pixel accuracy by enforcing consistency between boundary derived from the mask and boundary groundtruth.

## 2. Related work

**Semantic segmentation** Fully convolutional network (FCN) [8, 9] based methods [10, 11, 12, 13, 14, 15, 16, 17] make great progress in image semantic segmentation. In [10], last two downsample layers are removed to obtain dense prediction and dilated convolution operations are employed to enlarge the receptive field. DenseCRF [18, 10] is also used to capture long range dependencies between pixels. After that, an end-to-end CRF based method CRF-RNN is proposed to refine the semantic segmentation result. Unet [12], Deeplabv3+ [19], RefineNet [20] and DFN [21] adopt encoder-decoder structures that fuse the information in low-level and high-level layers to predict segmentation mask.

Recently, some works attempt to improve the feature representation ability through aggregating the contextual information. In [22], the ASPP module is used to capture contextual information by using different dilation convolutions. PSPNet [23] introduces pyramid pooling over sub-regions of four pyramid scales, and all the pixels within the same sub-region are treated as the context for the pixels belonging to the sub-region. ParseNet [16] utilizes global pooling to harvest context information for global representations. Zhao et al. [24] propose the pointwise spatial attention network which uses predicted attention map to guide contextual information collection.

**Semantic boundary detection** Recently, [3] extends the CNN based class-agnostic edge detector proposed in [25], and allows each edge pixel to be associated with more than one class. The CASENet [3] architecture combines low and high-level features with a multi-label loss function to supervise the fused activations. Most works use non-maximum-suppression (NMS) [7] as a postprocessing step in order to deal with the thickness of predicted boundaries. In [4], a simple and effective Thinning Layer and loss that can be used in conjunction with existing boundary detectors is proposed.

The most related works to this paper are Gated-SCNN [5] and BFP [6]. In [5], Towaki et al. adopt the two-stream architecture, including shape stream and classical mask stream, with gate design to facilitates the flow of information from the regular stream to the shape stream. Similarly, boundary is learned as an additional semantic class to enable the network to be aware of the boundary layout in [6]. Additionally, boundary aware feature propagation (BFP) module is proposed to harvest and propagate the local features within their regions isolated by the learned boundaries in the UAG-structured image. In this paper, we employ the semantic boundary to help refine semantic segmenta-

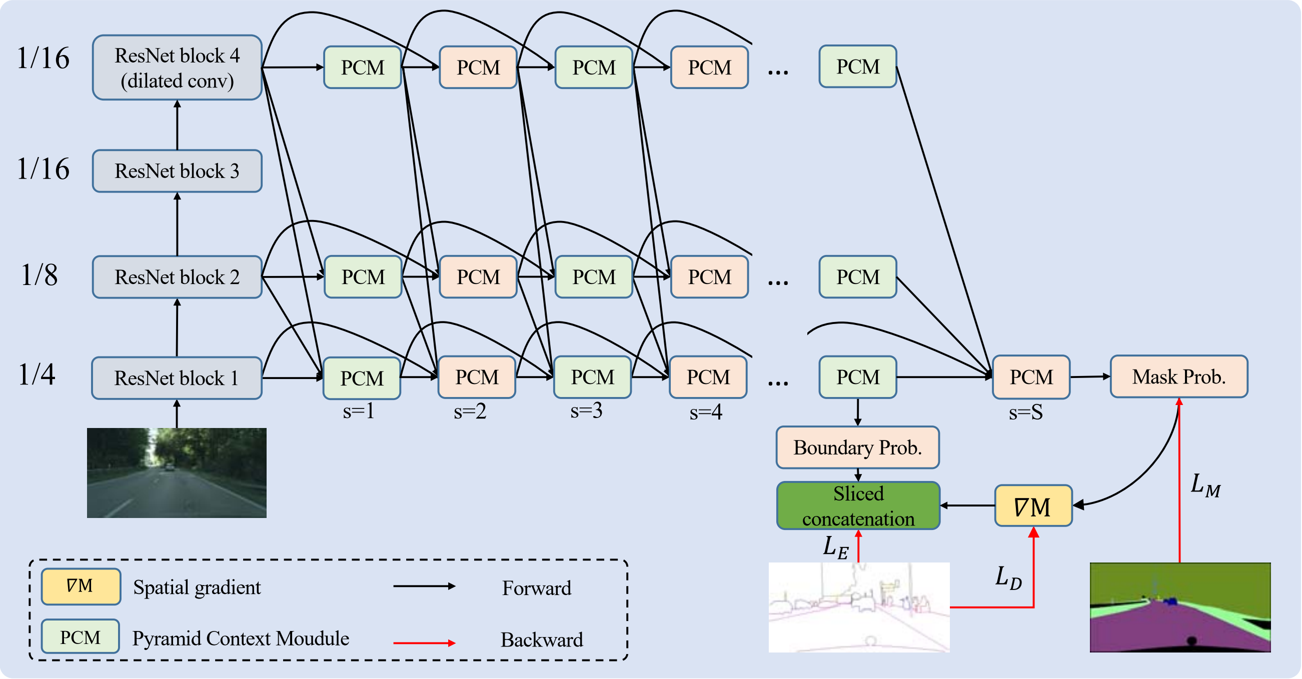

Figure 2. Overview of RPCNet for semantic segmentation and semantic boundary detection. The iterative pyramid context module (PCM) is used to interact between the two tasks. We leverage the context from one task to refine the feature map from another task alternately. The spatial gradient $\nabla M$ derived from semantic mask is fused together with the probability map from semantic boundary through sliced concatenation. The duality loss is also applied on semantic segmentation to improve the boundary accuracy of semantic mask.

tion results, which is much more consistent with semantic segmentation compared with binary boundary information. The two tasks are complementary to each other iteratively. In addition, we constrain semantic segmentation by using outer contour of semantic mask to compute duality loss, which improves boundary accuracy of semantic mask.

# 3. Approach

In this section, we present the proposed iterative pyramid context network (RPCNet) in detail, which is illustrated in Figure 2. We first give an overview of the whole architecture in section 3.1. Next, the iterative pyramid context module (PCM), which captures the pyramid context from multi-scale feature maps, is elaborated in section 3.2. Then we introduce details about the spatial gradient fusion in section 3.3. At last, we describe the computation process of duality loss in the framework in section 3.4.

## 3.1. Architecture

In this paper, we employ a pretrained residual network [26] with the dilated strategy [27] as the backbone. Note that we remove the downsampling operations and employ dilation convolutions in the last ResNet blocks, thus enlarging the size of the final feature map size to 1/16 of the input image. This retains more details without adding extra parameters. The feature maps with different scales from ResNet101 backbone are firstly fed into a $3 \times 3$ convolution followed by ReLU, batch normalization (BN) layer to reduce feature map number to 256, and the outputs are then taken as input of the iterative pyramid context module. We perform task-interaction by pyramid context refining on multiple levels. At each level, the two tasks are performed alternately. We leverage high-level feature maps, including the same level feature map, to refine the low-level feature maps.

More formally, let $s$ denote the step of pyramid context capturing ($1 \leq s \leq S$) as shown in Figure 2, and $F_s^t$ denotes $s$-th step feature map at $t$-th level. We use three level feature maps from the backbone. Thus, $t$ is from 0 to 2. $F_s^0$, $F_s^1$ and $F_s^2$ stand for feature maps with $\frac{1}{16}$, $\frac{1}{8}$ and $\frac{1}{4}$ size. When $s = 0$, $F_0^t$ denotes feature map extracted from the backbone.

After total $S$ steps, we can obtain fine feature maps for semantic segmentation and semantic boundary detection. For semantic boundary detection, we perform sliced concatenation operation on feature maps from semantic boundary detection task and feature maps containing semantic boundary information derived from semantic segmentation task. The spatial gradient is computed from the semantic segmentation task to obtain auxiliary semantic boundary probability map. For semantic segmentation, we adopt

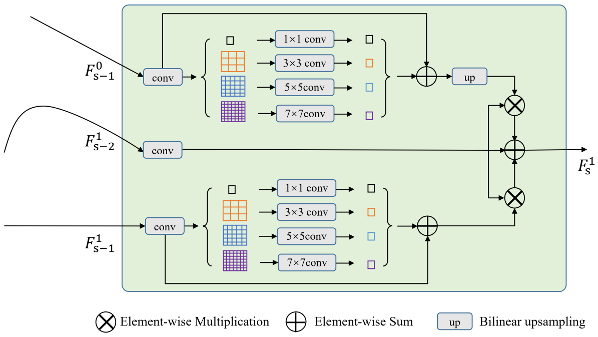

Figure 3. An example of pyramid context module (PCM) at step $s = 3$ and level $t = 1$. The feature maps from the higher level and the same level are firstly divided into multiple patches by global pooling. The global feature maps for different patches are then fed into a corresponding convolution layer to obtain the global context information. The feature map from one task (semantic segmentation or semantic boundary detection) is refined by these global context features. "up" operation uses bilinear interpolation method to upsample the input feature map into the same feature map size at level $t = 1$.

the duality loss to improve the boundary accuracy of semantic mask. As semantic boundary is easily derived from the semantic mask, we compute the spatial derivative $\nabla M$ on the probability map to obtain initial semantic boundary. The initial semantic boundary is compared with groundtruth boundary to compute duality loss.

### 3.2. Iterative Pyramid Context Module

We use the iterative pyramid context module to capture the global context from multiple levels to correlate the two tasks, where global context from one task is to purify the feature maps from another task.

For the input feature map $F_{s-2}^t$ at $s$-th step ($s \geq 2$) and $t$-level, we use higher level feature maps $F_{s-1}^{t'}$ ($t'$ is from 0 to $t$) at $s - 1$ step to refine it. When $s = 1$, we refine $F_0^t$ by $F_0^{t'}$. We perform context capturing by dividing the feature map into $G \times G$ patches and computing global context for each patch. Specifically, given the feature map $F_{s-1}^{t'} \in \mathbb{R}^{H\times W\times C}$, we obtain patch based context $P_{G\times G}^{t'} \in \mathbb{R}^{H\times W\times C}$ by

$$P_{G\times G}^{t'}(x,y) = \frac{1}{|S(x,y)|}\sum_{(h,w)\in S(x,y)} F(h,w) \quad (1)$$

where $x = 0, 1, ..., G-1$ and $y = 0, 1, ..., G-1$. $S(x, y)$ is used to denote a patch located at $x, y$. The features for each patch are then connected together by a $G \times G$ convolution layer, which yields a $C$ dimensional feature vector $F_G^{t'}$ as shown in Figure 3. We sum up all these context features and input feature map $F_{s-1}^{t'}$ to get context feature map $F_{context}^{t'}$. After the context feature map $F_{context}^{t'}$ is upsampled to the same size with $F_{s-2}^t$, we perform element-wise multiplication operation between them and sum up the refined feature map and input feature map $F_{s-2}^t$. Thus, we can obtain final output $F_s^t$ by

$$\begin{aligned} F_s^t &= F_{s-2}^t + \sum_{0\leq t'\leq t} F_{s-2}^t \odot F_{context}^{t'} \\ &= F_{s-2}^t + \sum_{0\leq t'\leq t} (F_{s-2}^t \odot F_{s-1}^{t'} + \sum_{G\in\{1,3,5,7\}} F_{s-2}^t \odot F_G^{t'}) \end{aligned} \quad (2)$$

where $\odot$ is element-wise multiplication and we empirically use the 4 different patch setup of $F_{s-1}^{t'}$ (i.e., $1\times1$, $3\times3$, $5\times5$ and $7\times7$ patches) to compute the set of feature maps $P_{G\times G}^{t'}$. A special case is $t = 0$, where there is no higher-level feature map used. This feature map would simply undergo a $3 \times 3$ convolution with ReLU and BN layer.

From Equation 2, we can see that the input feature map $F_{s-2}^t$ is refined by pyramid context representations, which are from different level feature maps with different scales and different context collection from different patch partition. The context refinement helps to yield finer feature representation on the one hand. On the other hand, it boosts the interaction between semantic segmentation and semantic boundary detection. We also use iterative pyramid context module to propagate contextual information between the two tasks, as shown in Figure 2. Thanks to the pyramid context module, the feature maps for the two tasks are closely correlated and mutually collaborate to boost the performance.

### 3.3. Spatial Gradient $\nabla M$ Fusion

After several steps of the pyramid context module, we can obtain the semantic mask probability map $M \in \mathbb{R}^{H\times W\times K}$ and the semantic boundary probability map $B \in \mathbb{R}^{H\times W\times K}$, where $K$ is the number of categories. We can obtain semantic boundary from semantic segmentation mask easily by spatial gradient deriving. Here, we use adaptive pooling to derive spatial gradient $\nabla M$, which is

$$\nabla M(x,y) = |M(x,y) - pool_k(M(x,y))| \quad (3)$$

where $x$ and $y$ denote the location of mask probability map and $|.|$ remarks the absolute value function. $pool_k$ is an adaptive average pooling operation with kernel size $k$. $k$ is used to control the derived boundary width and is set to 3 in the framework. Some examples are shown in Figure 4. For the input semantic segmentation mask, we can obtain precise semantic boundary results.

To augment the semantic boundary detection, we fuse the boundary probability map $B = \{B_1, B_2, ..., B_K\}$ and inferred boundary map $\nabla M = \{\nabla M_1, \nabla M_2, ..., \nabla M_K\}$ into new boundary $B'$ with $2K$ channels by sliced concatenation operation:

$$[B_1, \nabla M_1, B_2, \nabla M_2, ..., B_K, \nabla M_K]. \quad (4)$$

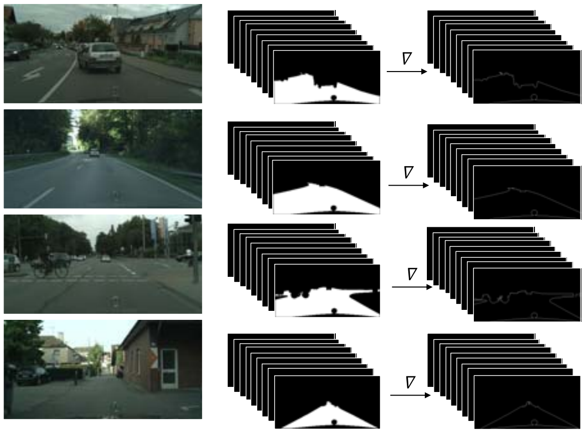

Figure 4. Some visualization examples for semantic mask and inferred semantic boundary through spatial gradient derivation on different categories (**best viewed in color**).

The resulting concatenated activation map is fed into a convolution layer with K-grouped convolution to produce a K-channel probability map $Y \in \mathbb{R}^{H \times W \times K}$, which is then used to compute semantic boundary detection loss with corresponding groundtruth.

Semantic boundary detection suffers from coarse boundary in previous works [3, 4] as the boundary is very sparse. In this paper, we fuse the inferred boundaries from the semantic mask to suppress non-edge pixels and localize detailed boundaries, which is desirable in semantic boundary detection.

### 3.4. Duality Loss

As shown in Figure 2, there are two tasks for our proposed network. We compute two kinds of loss corresponding to the probability feature maps for the tasks.

**Semantic Mask Loss** For semantic segmentation, it is common to compute cross-entropy loss for each pixel, which can be referred to as $L_M$. As it treats all pixels equally, the pixels around the boundary, which are ambiguous, are inconsistent with the groundtruth. We introduce a duality loss for semantic segmentation, assuming that there exists a consistency between the boundaries of segmented object with the groundtruth of object boundary.

As we derive semantic boundary $\nabla M$ from semantic mask, we measure inconsistency between it and semantic boundary groundtruth. We compute $l_1$ loss for it as follows

$$L_D = \sum_i |\nabla M_i - B_i^{gt}| \tag{5}$$

where $B^{gt}$ is the semantic boundary groundtruth derived from semantic segmentation mask groundtruth.

Two consistency constraints are imposed to enhance the performance of semantic segmentation. The cross-entropy loss term $L_M$ measures the consistency between the mask and its groundtruth. Additionally, the loss term $L_D$ measures the consistency between the derived boundary of semantic mask and semantic boundary groundtruth. Consequently, the loss function $L_m$ to measure the total error of semantic segmentation task is updated as:

$$L_m = L_M + \lambda_1 L_D \tag{6}$$

where $\lambda_1$ is a constant for balancing two losses.

**Semantic Boundary Loss** Compared with semantic segmentation, semantic boundary detection suffers more from the higher missing rate due to the sparsity of pixels around the boundary. In order to alleviate this impact, we follow [25, 3, 4] to define the following class-balanced cross-entropy loss function $L_2$:

$$L_E = -\sum_k^K \sum_i (\beta y_i^k log Y_i^k + (1-\beta)(1-y_i^k) log(1-Y_i^k)) \tag{7}$$

where $\beta$ is the percentage of non-edge pixels in the boundary groundtruth and $y_i^k$ is the groundtruth boundary label and binary indicating whether pixel $i$ belongs to class $k$.

Thus, the integrated loss function $L_{total}$ is finally formulated as below:

$$L_{total} = L_M + \lambda_1 L_D + \lambda_2 L_E \tag{8}$$

where $\lambda_2$ is a weight for balancing the boundary loss. In our experiments, we empirically set the two parameters $\lambda_1$ and $\lambda_2$ to 1 and 1000.

## 4. Experiments

### 4.1. Implementation details

The proposed RPCNet is implemented with PyTorch. The channel number of feature map in the pyramid context module is set to 256. In the training step, we adopt data augmentation similar to [28, 29]. The base learning rate is set to 0.001 for all our experiments. Momentum and weight decay coefficients are set to 0.9 and 0.0001, respectively. We train our model with Synchronized BN [30] with 6× NVIDIA 1080 Ti, and batch size is set to 6. Random crops and horizontal flip is also applied. Training input resolution is set to 894 × 894. We train the dataset with 180 epochs. We optimize the network by using the "poly" learning rate policy where the initial learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ with $power = 0.9$. In the ablation experiments, we set the batch size to 4 and training epochs to 60 in order to speed up the training procedure.

In the inference step, we follow [31, 32, 28] to crop an image into several parts by using a sliding window to keep consistent with the training process. Horizontal flipping is also adopted in the inference. We also use the multi-scale inference method with flipping when compared with state-of-the-art methods. In the ablation experiments, we

| Duality loss | $\nabla M$ | PCM | mIoU | MF (ODS) / AP |
|---|---|---|---|---|
| - | - | - | 78.14 | 73.61 / 72.81 |
| ✓ | - | - | 79.58 | 74.24 / 73.57 |
| ✓ | ✓ | - | 79.81 | 74.45 / 74.20 |
| ✓ | ✓ | $\{1\}$ | 79.92 | 74.65 / 74.29 |
| ✓ | ✓ | $\{1,3\}$ | 80.20 | 74.80 / 74.54 |
| ✓ | ✓ | $\{1,3,5,7\}$ | 80.43 | 75.54 / 75.14 |

Table 1. Ablation experiments for duality loss, $\nabla M$ fusion and pyramid context module (PCM). We set $S$ to 8 in the experiments.

| $S$ | mIoU | MF (ODS) / AP |
|---|---|---|
| 1 | 77.65 | - |
| 1 | - | 72.62 / 71.56 |
| 2 | 78.77 | 73.44 / 72.53 |
| 3 | 79.44 | 74.55 / 73.78 |
| 4 | 79.80 | 74.56 / 73.80 |
| 5 | 79.88 | 74.61 / 73.91 |
| 6 | 80.25 | 74.94 / 74.31 |
| 7 | 80.36 | 75.10 / 74.38 |
| 8 | **80.43** | **75.54 / 75.14** |

Table 2. Ablation experiments of iterative pyramid context module for semantic segmentation and semantic boundary. The iterative steps $S$ is set from 1 to 8. For $S = 1$, only one task is trained and evaluated.

use single-scale inference with flipping to evaluate. For semantic boundary inference, we also follow [4] to use Test-NMS as a post-processing method to generate a more sharp boundary.

## 4.2. Dataset

All of our experiments are conducted on the well-known Cityscapes dataset, which contains 2975 training, 500 validation and 1525 test images. Each image has a high resolution of 2048 × 1024 pixels with 19 semantic classes. Noted that no coarse data is employed in our experiments. We also follow [3, 4] to generate the ground truth boundaries for semantic boundary detection task.

## 4.3. Evaluation metric

In this work, we use the mean intersection of union (mIoU) [27, 24, 32] to evaluate the semantic segmentation task, which denotes the ratio of correctly classified pixels in a class over the union set of pixels predicted to this class and groundtruth, and then averaged over all classes, i.e., $\frac{1}{N}\frac{t_{ii}}{T_i+\sum_j t_{ji}-t_{ii}}$. Here, $N$ is the number of semantic classes, and $T_i$ is the total number of pixels in class $i$, while $t_{ij}$ indicates the number of pixels which belong to class $i$ and predicted to class $j$.

For semantic boundary detection, we follow the evaluation protocol proposed in [38, 4], which is considerably harder than the one used in [39, 3]. We report the maximum F-measure (MF) at optimal dataset scale (ODS), and average precision (AP) for each class. An essential parameter in the evaluation is the matching distance tolerance, which is defined as the maximum slack allowed for boundary predictions to be considered as correct matches to ground-truth. We follow [4] and set it to be 0.00375 in all our experiments.

| Method | Backbone | mIoU |
|---|---|---|
| DeeplabV2 [27] | ResNet101 | 70.4 |
| Piecewise [33] | ResNet101 | 71.6 |
| PSPNet [23] | ResNet101 | 78.8 |
| DeeplabV3+ [19] | ResNet101 | 78.8 |
| InPlaceABN [30] | WideResNet38 | 79.4 |
| GSCNN [5] | ResNet101 | 80.8 |
| DANet [32] | ResNet101 | 81.5 |
| RPCNet (SS + Flip) | ResNet101 | 81.8 |
| RPCNet (MS + Flip) | ResNet101 | **82.1** |

Table 3. Performance comparison between different strategies on Cityscape val set. "SS": single scale test. "MS": multi-scale test.

## 4.4. Ablation experiments

Our proposed method models iterative pyramid context to interact between semantic segmentation and semantic boundary detection. For semantic segmentation, we propose duality loss to ensure boundary consistency between the predicted mask and groundtruth. For semantic boundary detection, we fuse the spatial gradient $\nabla M$ from semantic mask into the semantic boundary probability map to suppress non-edge pixels. In order to verify the effect of the proposed components in this paper, we perform detailed ablation experiments to compare the performance after using or removing these modules. We summarize the results in Table 1 and Table 2.

**Duality Loss and $\nabla M$ Fusion** In Table 1, we first remove duality loss, spatial gradient $\nabla M$ fusion and pyramid context module to achieve mIoU score of 78.14% for semantic segmentation and MF / AP score of 73.61% / 72.81% for semantic boundary detection while $S$ is set to 8. Duality loss brings 1.44% improvement for semantic segmentation and 0.82% / 0.76% improvement for semantic boundary detection. The $\nabla M$ fusion also benefits the two tasks, which obtains mIoU score of 79.81% for semantic segmentation and MF (ODS) / AP score of 74.45% / 74.20% for semantic boundary detection.

**Pyramid Context Module** For the pyramid context module, we first compare different patch partition methods as shown in Table 1 and the patch partition setup ($1 \times 1$, $3 \times 3$, $5 \times 5$ and $7 \times 7$) achieves the best performance, which will be used in all other experiments. The pyramid context module boosts information exchange between semantic segmentation and semantic boundary detection. Compared with the

| Method | Backbone data | road | s.walk | build. | wall | fence | pole | t-light | t-sign | veg | terrain | sky | person | rider | car | truck | bus | train | motor | bike | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DeeplabV2 [27] | ResNet101 | 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 [20] | ResNet101 | 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 | 70.0 | 73.6 |
| PSPNet [23] | ResNet101 | 98.6 | 86.2 | 92.9 | 50.8 | 58.8 | 64.0 | 75.6 | 79.0 | 93.4 | 72.3 | 95.4 | 86.5 | 71.3 | 95.9 | 68.2 | 79.5 | 73.8 | 69.5 | 77.2 | 78.4 |
| AAF [34] | ResNet101 | 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 |
| DenseASPP [35] | DenseNet161 | 98.7 | 87.1 | 93.4 | 60.7 | 62.7 | 65.6 | 74.6 | 78.5 | 93.6 | 72.5 | 95.4 | 86.2 | 71.9 | 96.0 | 78.0 | 90.3 | 80.7 | 69.7 | 76.8 | 80.6 |
| PSANet [24] | ResNet101 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 80.1 |
| SeENet [36] | ResNet101 | 98.7 | 87.3 | 93.7 | 57.1 | 61.8 | 70.5 | 77.6 | 80.9 | 94.0 | 73.5 | 95.9 | 87.5 | 71.6 | 96.3 | 76.4 | 88.0 | 79.9 | 73.0 | 78.5 | 81.2 |
| ANNNet [37] | ResNet101 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 81.3 |
| CCNet [28] | ResNet101 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 81.4 |
| BFP [6] | ResNet101 | 98.7 | 87.0 | 93.5 | 59.8 | 63.4 | 68.9 | 76.8 | 80.9 | 93.7 | 72.8 | 95.5 | 87.0 | 72.1 | 96.0 | 77.6 | 89.0 | 86.9 | 69.2 | 77.6 | 81.4 |
| DANet [32] | ResNet101 | 98.6 | 87.1 | 93.5 | 56.1 | 63.3 | 69.7 | 77.3 | 81.3 | 93.9 | 72.9 | 95.7 | 87.3 | 72.9 | 96.2 | 76.8 | 89.4 | 86.5 | 72.2 | 78.2 | 81.5 |
| Ours | ResNet101 | 98.7 | 86.7 | 93.9 | 62.4 | 62.8 | 70.5 | 77.5 | 81.1 | 94.0 | 72.3 | 95.9 | 87.8 | 74.1 | 96.3 | 76.5 | 88.0 | 85.2 | 71.0 | 78.6 | **81.8** |

Table 4. Comparison vs state-of-the-art methods without coarse data training on the Cityscapes test set.

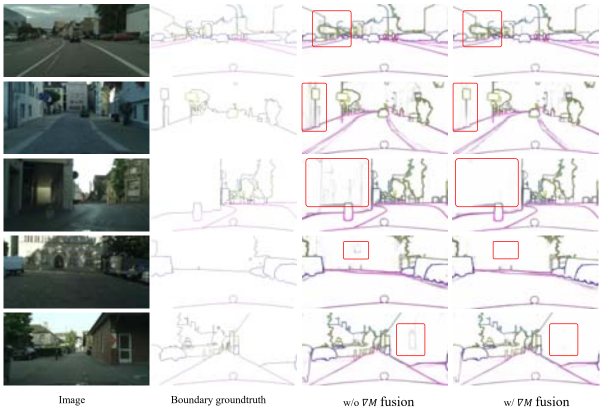

Figure 5. Some visualization comparison examples for semantic boundary detection with or without $\nabla M$ fusion (**best viewed in color**).

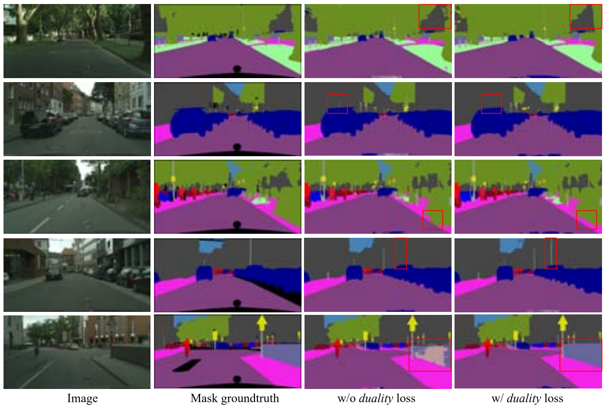

Figure 6. Some visualization comparison examples for semantic segmentation with or without duality loss used (**best viewed in color**).

setup without PCM, the PCM embedding can bring 0.62% improvement for semantic segmentation, 1.09% improvement on MF and 0.94% improvement on AP for semantic boundary detection.

**Iterative Pyramid Context Module** In Table 2, we first report the performance for single task setup ($S = 1$). It can be observed that combining two tasks together ($S = 2$) benefits both the two tasks. We also present the results for semantic segmentation and semantic boundary detection on different $S$. As the iterative pyramid context module can help to refine the feature maps with each other, we can see that increasing step number $S$ can bring constant improvement for both tasks. When $S = 8$, we can achieve mIoU score of 80.43% on semantic segmentation and MF / AP of 75.54% / 75.14% on semantic boundary detection. Thus, we set $S$ to 8 in our subsequent experiments.

### 4.5. Comparison with state-of-the-art works

**Semantic Segmentation** We first summarize state-of-the-art results on Cityscapes validation set in Table 3. We achieve the best performance compared with these methods. In particular, we outperform GSCNN by 1.3%, which leverages the binary edge as gate to boost the performance.

We also compare our method with state-of-the-art methods on Cityscapes test set. Specifically, we finetune our best model RPCNet with only fine annotated trainval data, and submit our test results to the official evaluation server. Results are shown in Table 4. We can see that our RPCNet achieves a new state-of-the-art performance of 81.8% on the test set. With the same backbone ResNet-101, our model outperforms DANet[32]. Besides, RPCNet also surpasses BFP [6], which makes use of binary edge information to propagate local features within their regions.

**Semantic Boundary Detection** We compare RPCNet with state-of-the-art methods for semantic boundary detection on Cityscapes validation set in Table 5. STEAL [4] proposes a novel loss to enforce the edge detector to predict a maximum response along the normal direction at edges, which is the current state-of-the-art method. However, we propose a better method to suppress non-semantic edges and attain high-quality representation. Our method achieves new state-of-the-art results over previous works by a large margin on both MF (ODS) metric and AP metric.

### 4.6. Visualization Results

To better understand the effect of the proposed methods, we present some visual examples. As shown in Figure 5, the visual examples for semantic boundary detection with or without $\nabla M$ are compared. We can see that after $\nabla M$

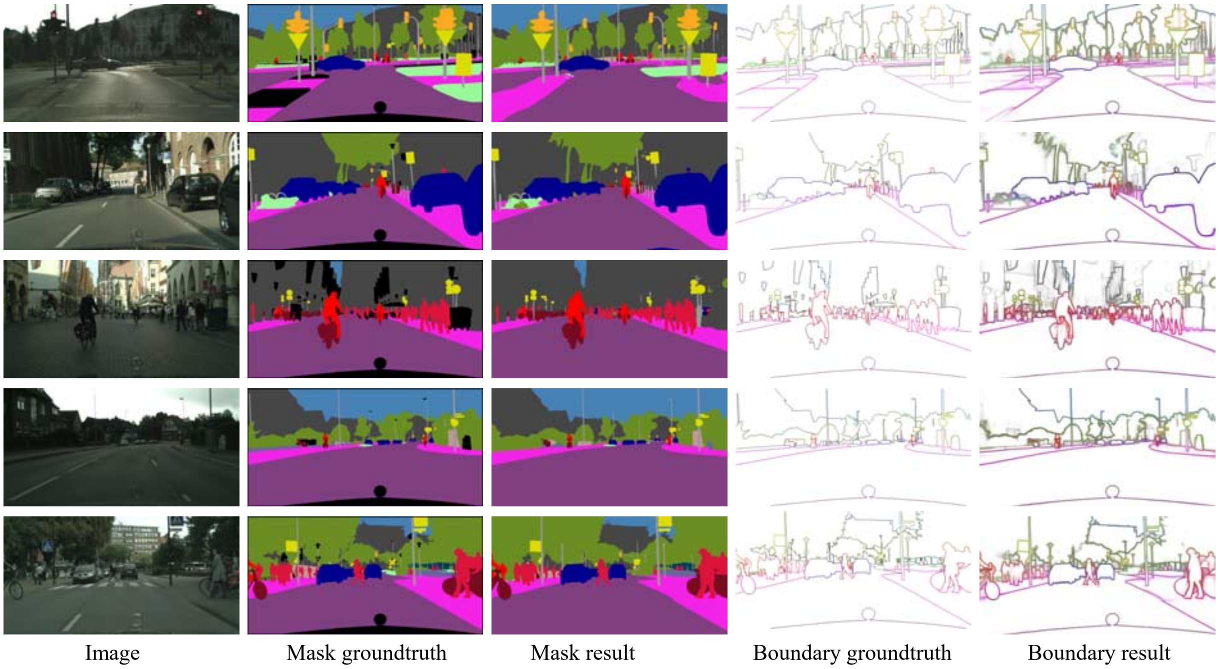

Figure 7. Visualization of results for semantic segmentation and semantic boundary detection (**best viewed in color**).

| **Metric** | Method | Test NMS | road | s.walk | build. | wall | fence | pole | t-light | t-sign | veg | terrain | sky | person | rider | car | truck | bus | train | motor | bike | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MF (ODS) | CASENet [3] | | 87.06 | 75.95 | 75.74 | 46.87 | 47.74 | 73.23 | 72.70 | 75.65 | 80.42 | 57.77 | 86.69 | 81.02 | 67.93 | 89.10 | 45.92 | 68.05 | 49.63 | 54.21 | 73.74 | 68.92 |
| | CASENet* [4] | | 87.23 | 76.08 | 75.73 | 47.86 | 47.57 | 73.67 | 71.77 | 75.19 | 80.58 | 58.39 | 86.78 | 81.00 | 68.18 | 89.31 | 48.99 | 67.82 | 50.84 | 55.30 | 74.16 | 69.29 |
| | CASENet* [4] | ✓ | 88.13 | 76.53 | 76.75 | 48.70 | 48.60 | 74.21 | 74.54 | 76.38 | 81.32 | 58.98 | 87.26 | 81.90 | 69.05 | 90.27 | 50.93 | 68.41 | 52.11 | 56.23 | 75.66 | 70.31 |
| | STEAL [4] | | 88.08 | 77.62 | 77.08 | 50.02 | 49.62 | 75.48 | 74.01 | 76.66 | 81.51 | 59.41 | 87.24 | 81.90 | 69.87 | 89.50 | 52.15 | 67.80 | 53.60 | 55.93 | 75.17 | 70.67 |
| | STEAL [4] | ✓ | 88.94 | 78.21 | 77.75 | 50.59 | 50.39 | 75.54 | 76.31 | 77.45 | 82.28 | 60.19 | 87.99 | 82.48 | 70.18 | 90.40 | 53.31 | 68.50 | 53.39 | 56.99 | 76.14 | 71.42 |
| | Ours | ✓ | **90.86** | **82.32** | **82.11** | **57.15** | **58.97** | **84.48** | **83.34** | **82.26** | **84.88** | **64.22** | **89.87** | **86.28** | **78.47** | **92.61** | **67.75** | **82.79** | **68.48** | **69.20** | **80.09** | **78.22** |
| AP | CASENet [3] | | 54.58 | 65.44 | 67.75 | 37.97 | 39.93 | 57.28 | 64.65 | 69.38 | 71.27 | 50.28 | 73.99 | 72.56 | 59.92 | 66.84 | 35.91 | 56.04 | 41.19 | 46.88 | 63.54 | 57.65 |
| | CASENet* [4] | | 68.38 | 69.61 | 70.28 | 40.00 | 39.26 | 61.74 | 62.74 | 73.02 | 72.77 | 50.91 | 80.72 | 76.06 | 60.49 | 79.43 | 40.86 | 62.27 | 42.87 | 48.84 | 64.42 | 61.30 |
| | CASENet* [4] | ✓ | 88.83 | 73.94 | 76.86 | 42.06 | 41.75 | 69.81 | 74.50 | 76.98 | 79.67 | 56.48 | 87.73 | 83.21 | 68.10 | 91.20 | 44.17 | 66.69 | 44.77 | 52.04 | 75.65 | 68.13 |
| | STEAL [4] | | 89.54 | 75.72 | 74.95 | 42.72 | 41.53 | 65.86 | 67.55 | 75.84 | 77.85 | 52.72 | 82.70 | 79.89 | 62.59 | 91.07 | 45.26 | 67.73 | 47.08 | 50.91 | 70.78 | 66.44 |
| | STEAL [4] | ✓ | 90.86 | 78.94 | 77.36 | 43.01 | 42.33 | 71.13 | 75.57 | 77.60 | 81.60 | 56.98 | 87.30 | 83.21 | 66.79 | 91.59 | 45.33 | 66.64 | 46.25 | 52.07 | 74.41 | 68.89 |
| | Ours | ✓ | **91.27** | **83.87** | **84.00** | **53.18** | **54.96** | **84.55** | **85.48** | **84.66** | **86.15** | **61.18** | **90.72** | **88.95** | **79.95** | **94.40** | **68.11** | **85.47** | **68.53** | **69.44** | **82.17** | **78.79** |

Table 5. Quantitative results on the val set on the Cityscapes dataset. We use ResNet101 pretrained on ImageNet as backbone. CASENet* is the reimplementation of CASENet in [4]. Scores are measured by %.

fusion, we can obtain more accurate semantic boundaries. The pixels, even though they belong to edges but not semantic boundaries, are suppressed so that we can locate pixels belonging to semantic boundaries. We also visualize the examples for semantic segmentation with or without duality loss in Figure 6. It can be observed that the novel loss can help to generate more precise boundaries. The misclassified "Pole" is also recognized after duality loss is used. At last, we also visualize some examples for both semantic segmentation and semantic boundary detection. In Figure 7, the mask outputs are accurate and the quality of the semantic boundaries is also very high.

## 5. Conclusion

In this paper, we have presented a joint-task framework for both semantic segmentation and semantic boundary detection. We use an iterative pyramid context from one task at multiple scales to refine the feature map of another task alternately, which helps the two tasks interact with each other. In order to resolve the sparse boundary issues, we fuse derived boundary from semantic segmentation mask into semantic boundary probability map to suppress non-semantic edge pixels. The novel loss function originated from the dual constraint is designed to improve further the performance for semantic segmentation, which ensures the consistency between semantic mask boundary and boundary groundtruth. The comprehensive experiments on the Cityscapes dataset verify the effectiveness of the proposed framework and show that proposed RPCNet outperforms current state-of-the-art works not only on semantic segmentation task but also on semantic boundary detection task.

## 6. Acknowledgments

This work is supported by Hong Kong RGC GRF 16206819, Hong Kong RGC GRF 16203518, Hong Kong T22-603/15N.

## References

[1] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018. 1

[2] Zhang-Wei Hong, Chen Yu-Ming, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, Hsuan-Kung Yang, Brian Hsi-Lin Ho, Chih-Chieh Tu, Yueh-Chuan Chang, Tsu-Ching Hsiao, et al. Virtual-to-real: Learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*, 2018. 1

[3] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *CVPR*, 2017. 1, 2, 5, 6, 8

[4] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *CVPR*, 2019. 1, 2, 5, 6, 7, 8

[5] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *CVPR*, 2019. 1, 2, 6

[6] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. *ICCV*, 2019. 1, 2, 7

[7] John Canny. A computational approach to edge detection. *TPAMI*, 1986. 2

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[9] E Shelhamer, J. Long, and T Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2017. 2

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015. 2

[11] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCI*, 2015. 2

[13] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 2

[14] Md Amirul Islam, Mrigank Rochan, Neil DB Bruce, and Yang Wang. Gated feedback refinement network for dense image labeling. In *CVPR*, 2017. 2

[15] Jun Fu, Jing Liu, Yuhang Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *arXiv preprint arXiv:1708.04943*, 2017. 2

[16] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *ICLR*, 2016. 2

[17] Mingmin Zhen, Jinglu Wang, Lei Zhou, Tian Fang, and Long Quan. Learning fully dense neural networks for image semantic segmentation. In *AAAI*, 2019. 2

[18] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 2

[19] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. 2018. 2, 6

[20] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *CVPR*, 2017. 2, 7

[21] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018. 2

[22] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[23] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 6, 7

[24] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 2, 6, 7

[25] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2, 5

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2016. 3, 6, 7

[28] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *ICCV*, 2019. 5, 7

[29] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 5

[30] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018. 5, 6

[31] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 5

[32] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *CVPR*, 2019. 5, 6, 7

[33] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 6

[34] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, 2018. 7

[35] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018. 7

[36] Pang Yanwei, Li Yazhao, Jianbing Shen, and Ling Shao. Towards bridging semantic gap to improve semantic segmentation. *ICCV*, 2019. 7

[37] Zhen Zhu, Mengdu Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. *ICCV*, 2019. 7

[38] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, BVK Vijaya Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *ECCV*, 2018. 6

[39] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6