Towards Bridging Semantic Gap to Improve Semantic Segmentation

Yanwei Pang[1], Yazhao Li[1], Jianbing Shen[2]*, Ling Shao[2]
[1]Tianjin University, Tianjin, China [2]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
pyw@tju.edu.cn, lyztju@tju.edu.cn, shenjianbingcg@gmail.com, ling.shao@ieee.org

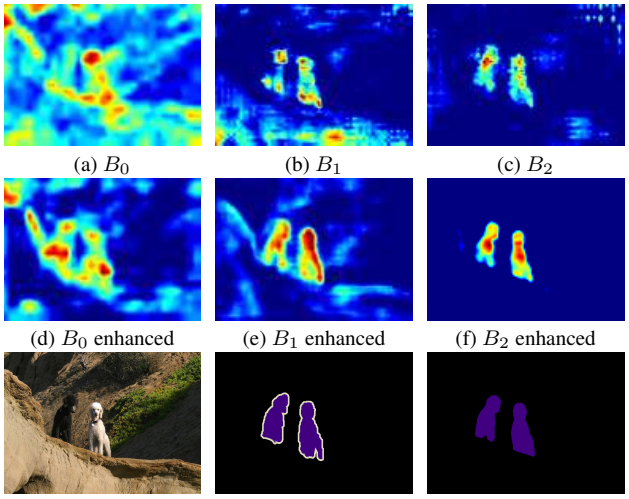
Abstract

*Aggregating multi-level features is essential for capturing multi-scale context information for precise scene semantic segmentation. However, the improvement by directly fusing shallow features and deep features becomes limited as the semantic gap between them increases. To solve this problem, we explore two strategies for robust feature fusion. One is enhancing shallow features using a semantic enhancement module (SeEM) to alleviate the semantic gap between shallow features and deep features. The other strategy is feature attention, which involves discovering complementary information (*i.e.*, boundary information) from low-level features to enhance high-level features for precise segmentation. By embedding these two strategies, we construct a parallel feature pyramid towards improving multi-level feature fusion. A Semantic Enhanced Network called SeENet is constructed with the parallel pyramid to implement precise segmentation. Experiments on three benchmark datasets demonstrate the effectiveness of our method for robust multi-level feature aggregation. As a result, our SeENet has achieved better performance than other state-of-the-art methods for semantic segmentation.*

1. Introduction

Scene semantic segmentation is a high-level visual task whose goal is to assign a corresponding semantic label to each pixel in an image. To deal with complex scale variations, it is essential to extract multi-scale robust features and abundant context information [21, 57, 19, 47].

State-of-the-art semantic segmentation methods are typically based on the Fully Convolutional Network (FCN) [36, 53]. Most FCN-based methods [48] tend to construct an encoder branch to gradually improve the semantic levels without using fully connected layers. To restore the resolution information, a cascaded decoder stream is widely explored [2, 41, 5, 49]. A skip connection [38, 30, 14] is frequently used to combine the encoder and decoder features. The decoder stream, which acts as a feature pyramid, helps aggregate multi-level features and capture multi-scale context information. In this framework, features from shallow layers encode more detailed information, while features from deep layers encode more semantic information to distinguish different classes. The shallow features and deep features are complementary for precise segmentation [59].



(a) B_0 (b) B_1 (c) B_2
(d) B_0 enhanced (e) B_1 enhanced (f) B_2 enhanced
(g) Image (h) Ground truth (i) SeENet result

Figure 1. Feature enhancement towards bridging the semantic gap. (a)∼(c) show the features of (g) from the final three blocks respectively. A semantic gap exists when combining them. (d)∼(f) show features after enhancing (a)∼(c) respectively. Combining (d) with (b) is more robust than combining (a) and (b). (h) shows the segmentation ground truth, and (i) shows the segmentation result by our method.

However, shallow features are low-level for the edges, lines, and corners, while deep features are high-level for measuring object characteristics, classification, and scene parsing. We refer this kind of difference of semantic granularity as semantic gap in this paper. This gap exists to disturb the multi-level feature fusion but has rarely been explored. We show some feature heat maps in Fig. 1 (a)∼(c) of an image (g). These features are generated from the last three stages of a FCN network to present the gap existed in different layers. As can be seen, features in (a), which are from the shallow layer, contain more local region and edge information for the dogs and the background hill. Fea-

*Corresponding author: Jianbing Shen

tures in (c) from the deep layer contain more discriminative information for the segmentation. That is, features in shallow layers have coarse information, while features in deep layers embed more representative information. It has been observed that a semantic gap exists between these features. Introducing these shallow features to deep features will bring some background 'noisy' features, which influences the feature robustness and may cause semantic inconsistency. Thus, directly fusing these shallow and deep features is less effective.

Based on this observation, we argue that the semantic gap should be considered when aggregating multi-level features. One solution can be found to promote robust feature aggregation is fusing the features which have less semantic distance. Thus, we propose to aggregate the shallow features to deep features gradually in a bottom-up manner. Further enhancing the shallow features helps alleviate the gap in the neighboring pyramid features. Besides, eliminating the 'noisy' features in shallow features and discovering the complementary information helps feature aggregation without impairing the high-level features. With these strategies, we propose a robust feature fusion method towards bridging the semantic gap and improving the segmentation performance.

The contribution and characteristics of the proposed method are as follows: *A parallel pyramid method, which is presented in a bottom-up manner, is proposed to bridge the semantic gap and implement robust multi-level feature aggregation. Two strategies are proposed for feature fusion. One strategy involves improving the shallow features by introducing a semantic enhancement module. The other strategy consists of extracting complementary information from very shallow features by designing a reversed boundary attention module for boundary refinement. Subsequently, a Semantic Enhanced Network, SeENet, is constructed by embedding the proposed parallel feature pyramid for semantic segmentation. As a result, SeENet achieves the top performance on several benchmark datasets.*

2. Related Work

Fully Convolutional Networks have been widely explored to improve segmentation performance [37, 30]. It has been shown extracting multi-scale context information and enhancing feature discriminability [59, 52, 13, 15, 39] are beneficial for dealing with complex scale variations and implementing precise segmentation. In this work, we discuss the networks with typical modules that exploit multi-scale information and methods for feature enhancement.

Top-down feature pyramid. A top-down feature pyramid [32] is aimed at producing multi-resolution features from different stages of the network and fusing these features from top to bottom, gradually. The decoder stream in the encoder-decoder method works as a top-down pyramid module. We show this kind of pyramid method in Fig. 2 (a). SegNet [2] adopted an encoder-decoder network and used pooling indices to restore high resolution. In DeconvNet [38], deconvolution and unpooling were used for the decoder stream to enhance the resolution. U-Net [41] was proposed to build skip connections from the encoder features to the corresponding features in the decoder stream and implement multi-level feature aggregation. Many feature pyramid methods with skip connection [59, 40, 27, 20, 26] have since been explored for semantic segmentation, as well as other tasks [18, 32], and have achieved great improvements. RefineNet [30] also explored multi-level features for high-resolution scene segmentation by multi-path refinement and demonstrated effectiveness.

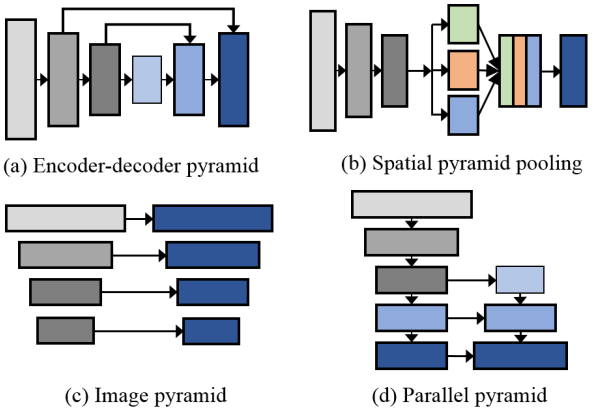


Figure 2. Different pyramids for capturing multi-scale features.

Spatial pyramid pooling. A spatial pyramid pooling module [23], as shown in Fig. 2 (b), typically contains several pooling branches to generate multi-scale features. PSPNet [60] was proposed to perform spatial pooling at several grid scales to capture multi-scale information. DeepLabv2 [8] proposed atrous spatial pyramid pooling (ASPP), where parallel atrous convolutional layers with different rates of filters capture multi-scale information [9]. The ASPP module was improved in DeepLabv3 [9] by integrating a global pooling branch. DenseASPP [52] developed a more effective module by connecting different atrous convolution in a dense manner to cover a larger range of scales. All of the ASPP variants [9, 10, 52, 58, 6, 42] were stacked at the top of their backbone networks for prediction.

Image pyramid. By using an image pyramid, as shown in Fig. 2 (c), one image is resized to different scales and input to the network. Eigen *et al*. [16] proposed a multi-scale network to progressively refine the output. Lin *et al*. [31] adopted multi-scale inputs and fused the features. Liu *et al*. [34] proposed to use multi-scale patches and aggregate the results. Although using multi-scale inputs helps extract abundant features, image pyramid based methods are computationally expensive and consume large GPU memory [13], which limits their practical application.

Feature enhancement. To enhance features, enlarging

receptive fields and capturing more context information have been explored. To achieve this, Peng *et al*. [40] proposed to capture more global context information using a large kernel convolution in the top of the network. More recently, atrous convolution has been widely explored [7, 54, 50, 55, 46, 4, 42, 43] for capturing context information. Zhang *et al*. proposed to improve semantic levels by introducing multi-stage semantic supervision [59]. Inception modules [44] have also been explored [33] to enhance feature discriminability and robustness.

Different from previous pyramid methods for semantic segmentation, we construct a parallel pyramid, as shown in Fig. 2 (d), towards bridging the semantic gap between multi-level features and aggregating multi-level features robustly. To achieve this, multi-level features are fused progressively in a bottom-up pyramid to shrink the feature distance. In the parallel pyramid, we introduce semantic enhancement modules to enhance the shallow features, and use an attention module to discover the complementary information and enhance the deep features.

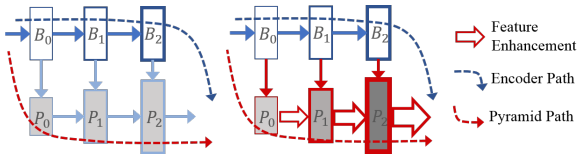
3. Parallel Pyramid for Robust Feature Fusion

In this work, we propose a parallel pyramid for robust multi-level feature aggregation. In our parallel pyramid, two strategies are adopted to alleviate the semantic inconsistency. One is enhancing the discriminability of shallow features. The other is only capturing complementary information for deep layers. We first present our parallel pyramid. Then, we present the details of these two strategies in our method.

3.1. Parallel Feature Pyramid

Multi-level feature aggregation. Capturing multi-scale features is essential for resolving complex scale variations in scene segmentation. Features from different stages of the network contain information of various scales. Different from previous pyramid methods, we propose a new parallel pyramid method to progressively fuse the features from neighboring stages in a bottom-up manner. We show the architecture of the pyramid in Fig. 3 (a). The parallel pyramid is built to aggregate multi-level features (B_0, B_1, and B_2) from the backbone and enhance the multi-scale information. Note that we illustrate one typical parallel pyramid with the backbone features of B_0, B_1, and B_2, which are given the same resolution by using the dilation strategy as [56, 8]. The pyramid is constructed by fusing the features from bottom to top progressively. As a result, multi-level features are aggregated to enrich the multi-scale information.

Dual-path aggregation. As shown in Fig. 3, the parallel feature pyramid acts as an efficient decoder stream. Two different feature-extraction paths exist in our method. One path, found in the backbone encoder stream, consists of several convolutional blocks cascaded to gradually improve semantic information. The other path consists of a feature pyramid that eases information flow from the bottom to top layers and hierarchically fuses multi-level features. Incorporating dual-path information will promote feature fusion as well as resolve large-scale variations in complex scenes.



(a) Basic parallel pyramid (b) Semantic-enhanced prallel pyramid

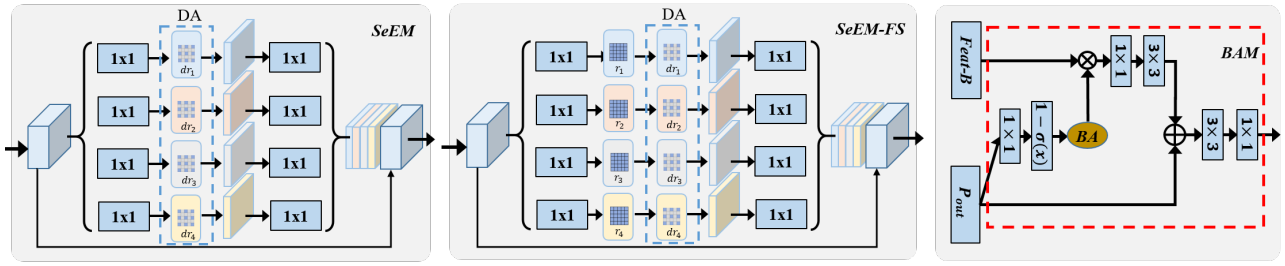
Figure 3. Parallel feature pyramid. B_i represents features from the backbone layer i. P_i represents features in the pyramid layer i. Bolder contour lines mean with higher semantics.

Shortening feature distance. In the networks with encoder-decoder architecture, feature pyramid is typically constructed in a cascaded manner. Skip connections are employed between encoder and decoder features. Although traditional encoder-decoder based top-down pyramid methods have achieved great success, the semantic gap existing between the shallow layers and deep layers limits the performance of feature fusion [59]. To aggregate multi-level features robustly, this semantic gap should be considered and alleviated. Therefore, We construct our feature pyramid in a parallel manner and hierarchically fuse the features of the encoder stream, which can shorten the feature distance as compared to the cascaded pyramid.

To further alleviate the semantic gap between multi-level features, we propose two strategies for improving feature fusion. One strategy involves enhancing the features of shallow layers using a Semantic Enhancement Module (SeEM) before fusing the shallow and deep features. As shown in Fig. 3 (b), we first enhance the shallow features of B_0 to P_0, to capture more context information similar with B_1 and alleviate the feature inconsistency. The other strategy consists of extracting complementary information instead of using all the shallow features when fusing with high-level features. As known, some boundary information usually exists in very shallow layers, which is helpful for enhancing the deep features for precise segmentation. However, the semantic gap is much larger for these features. Based on this strategy, we construct a Boundary Attention Module (BAM) to extract boundary information. We present the details of the SeEM and BAM as follows.

3.2. SeEM for Feature Enhancement

To bridge the semantic gap in feature fusion, we propose to enhance the shallow features by a semantic enhancement module. Enlarging the receptive fields and capturing more context information can help improve the representation ability of features. As discussed in Sec. 2, we can use ASPP, an inception module, or a large kernel method to enhance the shallow features. Considering computational



(a) Semantic Enhancement Module (b) Semantic Enhancement Module with full sampling (c) Boundary Attention Module

Figure 4. Semantic Modules in the proposed parallel pyramid method for improving feature fusion. We introduce semantic enhancement modules (a) and (b) to enhance the semantics of shallow features, and propose a boundary attention module (c) to extract complementary information from very shallow features and enhance the deep features. 'DA' represents depthwise atrous convolution. 'dr_i' represents the dilation rate. 'r_i' represents the kernel size of convolutional layer. 'BA' represents boundary attention.

complexity, we implement our SeEM based on depthwise ASPP [10]. Different from the ASPP in [10] which is used at the top of the network for the final prediction, the designed Semantic Depthwise ASPP (S-DASPP) is used to enhance shallow features and ensure robust feature fusion. Besides, because the SeEM is used inside the feature pyramid instead of for the final prediction, using very large dilation rates, such as those in [10] (*i.e.*, (6, 12, 18)), is not suitable for shallow layers. Thus, we adopt relatively small dilation rates (*e.g.*, (1, 2, 4, 8)) in the shallow pyramid layer. We do not use the 'image pooling' in SeEM but introduce a residual connection to make the learning procedure stable. Note that, using this kind of ASPP for constructing the SeEM is not our innovation. We can also embed a large kernel or inception modules in our pyramid to enhance semantics. In addition, we propose a new SeEM with full sampling to make precise segmentation. We show the detailed implementation below.

Semantic Depthwise ASPP. Considering both efficiency and effectiveness, the SeEM is constructed with an S-DASPP module as shown in Fig. 4(a) for capturing multi-scale context information and feature enhancement. S-DASPP is composed of four parallel depthwise atrous convolution branches. In each branch, we first use a 1×1 convolutional layer to reduce the input channel number to a small value (*e.g.*, 128). Then, a depthwise atrous convolutional layer is used to enlarge the receptive field, which is followed by another 1×1 convolutional layer to fuse the channel information. Batch normalization [25] and a ReLU activation are used for each convolutional layer. We concatenate the output of the four branches with the input features to ease the network training with a denser connection manner. One kernel can be represented as $w^{m \times n \times c}$ in the atrous convolutional layer and $w^{m \times n \times 1}$ in the depthwise atrous convolutional layer, where $m \times n$ are 3×3 in our S-DASPP and c represents the channel number of the input. Obviously, less parameters are needed for a depthwise convolution. To capture multi-scale information, different dilation rates (dr_1, dr_2, dr_3, dr_4) are adopted in the four branches. To improve the semantics of shallow features, we use dilation rates $(1, 2, 4, 8)$ to configure SeEM. By using S-DASPP, multi-scale context information can be further enriched by the multi-scale atrous convolution branches.

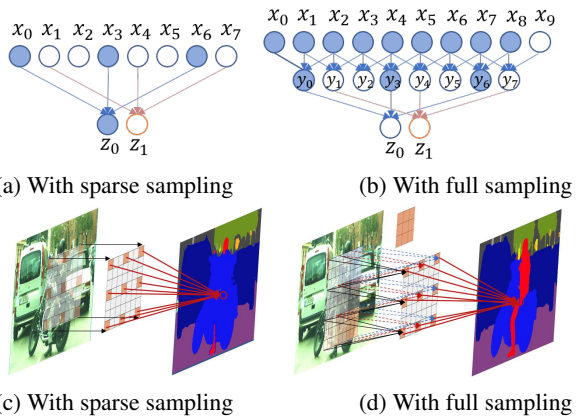
S-DASPP with full sampling. Atrous convolution is typically used to capture large receptive fields in a sparse sampling manner. To generate precise segmentation results, we propose to construct a semantic enhancement module with a full sampling component (SeEM-FS), as shown in Fig. 4(b). After the 1×1 convolutional layer in each branch for channel reduction, we first employ a convolutional layer with $r_i \times r_i$ filters to capture the local region information. The following depthwise atrous convolutional layer with dilation rate dr_i is able to capture global context information in a full sampling manner. We illustrate the sparse sampling and full sampling methods in Fig. 5. In one dimension, as (a), denoting x_i as the input, for an atrous convolution with a dilation rate $r = 3$, the output z_0 can be represented as:

$$z_0 = f_{ac}(x_0, x_3, x_6), \tag{1}$$

where $f_{ac}(x)$ represents the atrous convolution operation. Thus, z_0 can only capture information from $\{x_0, x_3, x_6\}$ in a sparse sampling manner. We demonstrate the full sampling method in Fig. 5(b). Using a convolution kernel $r \times 1$, feature y_i can capture local information from (x_i, x_{i+1}, x_{i+2}). Then, z_0 can be formulated as:

$$\begin{aligned} z_0 &= f_{ac}(y_0, y_3, y_6) \\ &= f_{ac}(f_c(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)), \end{aligned} \tag{2}$$

where $f_c(x)$ represents the convolution operation. Thus, z_0 covers all input points ranging from x_0 to x_8 (*i.e.*, $\{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$) in a full sampling manner. We present one example in Fig. 5 (c) and (d) to demonstrate the advantages of full sampling. The rider on the motorbike in (c) is wrongly segmented to motorbike due to the use of sparse sampling (most sampling points are from the motorbike). By using the full sampling method, the rider can be correctly segmented as in (d). To avoid significantly increasing computational cost, we use $\{r \times 1, 1 \times r\}$ and



(a) With sparse sampling (b) With full sampling

(c) With sparse sampling (d) With full sampling

Figure 5. Atrous convolution with sparse sampling in SeEM and full sampling in SeEM-FS.

$\{1 \times r, r \times 1\}$ convolution groups to approximate the $r \times r$ convolution, which were developed by [44, 40].

3.3. Boundary Attention Module

While cascaded convolutional operations can enlarge receptive fields and capture more global context information, they also result in a loss of boundary information. To restore boundary information, some methods directly add or concatenate features from very shallow layers with deep layers. However, the features in shallow layers not only contain boundary information but also contain texture information inside objects or stuff, which may negatively effect the robustness of high-level features. Thus, to fuse the very shallow features and deep features, we propose to remove the redundant features and keep the boundary information in shallow features when combining with deep features. To achieve this, a boundary attention module (BAM), as shown in Fig. 4(c), is designed and employed in our pyramid.

Inspired by the reverse attention mechanism [11], we propose to extract boundary information by paying attention to the regions that are not salient in high-level features (*i.e.*, P_{out}). We denote the features in P_{out} as $P \in \mathbb{R}^{h \times w \times c_p}$ and denote *Feat-B* as $B \in \mathbb{R}^{h \times w \times c}$. We first apply a 1×1 convolution on P and generate $\hat{P} \in \mathbb{R}^{h \times w \times c}$. The boundary attention is generated as:

$$A = 1 - \sigma(\hat{P}) = 1 - \frac{1}{1 + e^{-\hat{P}}}. \tag{3}$$

Then, the boundary features $\hat{B} \in \mathbb{R}^{h \times w \times c}$ is captured as:

$$\hat{B} = A \odot B, \tag{4}$$

where $\odot$ represents the Hadamard product. We further fuse the boundary features using 1×1 and 3×3 convolutional layers. By the proposed BAM, the generated boundary features and the original high-level features are complementary. Finally, we use one 3×3 convolutional layer for feature fusion and another 1×1 convolutional layer to generate the final segmentation results.

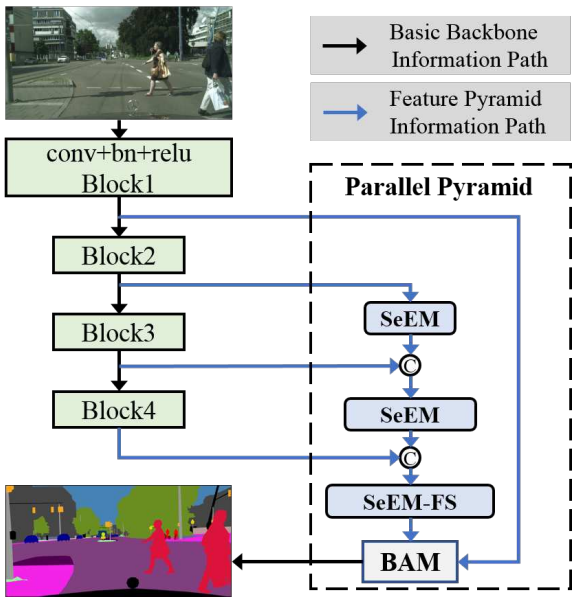


Figure 6. Overview of our SeENet for scene segmentation. There are two information paths. One is the basic backbone information path. The other is our feature pyramid information path in parallel for multi-level features aggregation.

3.4. SeENet for Segmentation

With the proposed parallel feature pyramid, we build a deep FCN network, Semantic Enhanced Network (SeENet), with a pre-trained ResNet [24] backbone for semantic segmentation. We show the main architecture of SeENet in Fig. 6. We follow the prior work of [56, 8], removing the subsampling layers in Block3 and Block4 and using a dilated strategy on the pretrained backbone at Block3 and Block4. Our parallel pyramid is built upon the output of Block1∼Block4. Specifically, we first enhance the shallow features from Block2 using an SeEM module. Then, we fuse the enhanced features of Block2 with features of Block3 through a concatenation operation. We progressively fuse the features of Block2∼Block4 with our SeEM. Before performing the final prediction, we apply an SeEM-FS to further enhance the semantics and capture abundant context information. The dilation rates in the three SeEMs are $(1, 2, 4, 8)$, $(3, 6, 9, 12)$, and $(7, 13, 19, 25)$. The channel number for the 1×1 convolutional layer in SeEM and SeEM-FS is set as 128. To refine the boundary information, we introduce the very shallow features from Block1, which contain much detailed information, to our BAM module.

4. Experimental Results

To demonstrate the effectiveness of the proposed method, experiments are conducted on three public segmentation datasets: Pascal VOC 2012 [17], CamVid [3], and Cityscapes [12]. Ablation experiments are carried out on the Pascal VOC 2012 dataset to evaluate the contribution of

Methods	R50	R101	PFP	SeEM	SeEM-FS	BF	BAM	mIoU (%)
(a) Baseline	✓							70.9
(b) +PFP	✓		✓					72.7
(c) Baseline		✓						74.9
(d) +PFP		✓	✓					75.7
(e) +SeEM-FS		✓			✓			75.9
(f) +PFP^a		✓	✓	✓				76.9
(g) +PFP^b		✓	✓		✓			77.5
(h) +PFP^c		✓	✓	✓	✓			77.4
(i) +PFP^c +BF		✓	✓	✓	✓	✓		77.7
(j) SeENet		✓	✓	✓	✓		✓	78.1

PFP^a embeds three SeEM modules. PFP^b embeds three SeEM-FS modules. PFP^c embeds two SeEM modules and one SeEM-FS module.

Table 1. Ablation studies for each part of SeENet on the Pascal VOC 2012 *validation* set. 'R50' and 'R101' represent two backbones with ResNet50 and ResNet101 [24]. 'PFP' represents the proposed Parallel Feature Pyramid without embedding SeEM and BAM. 'BF' means boundary refinement by skip connection.

each part of our parallel feature pyramid.

Our experiments are conducted using Tensorflow [1]. Following previous work [8, 56], we use iter learning rate scheduling (*i.e.*, $lr = baselr * (1 - \frac{iter}{total\_iter})^{power}$ with $power = 0.9$) to train the networks. We set $baselr = 0.001$ for the VOC 2012 and Camvid datasets, and $baselr = 0.007$ for the Cityscapes dataset. We set the weight decay to 0.0001 and momentum to 0.9. We augment data by carrying out random left-right flips, randomly scaling the image $(0.5, 2.0)$, randomly cropping and zero padding for training. We use the standard Cross Entropy loss to supervise the model training.

4.1. Results on Pascal VOC 2012 Dataset

Pascal VOC 2012 is a benchmark dataset containing 20 foreground classes and one background class (making a total of 21 classes) for semantic segmentation. As in [60, 56, 9], we use the extra annotation [22] along with the original dataset to construct the *training* set (10582 images), *validation* set (1449 images), and *test* set (1456 images). We first conduct ablation experiments to evaluate each part of our SeENet on the *validation* set, and then compare them with other state-of-the-art methods on the *test* set by submitting the results to the Pascal VOC server. The performance is measured in terms of mean pixel intersection-over-union (mIoU), averaged across the 21 classes.

4.1.1 Ablation Studies

For ablation experiments, we train all the models on the *training* set, and evaluate them on the *validation* set. We train the network with a small crop-size of 320×320 and a mini-batch of 10 for 50K iterations. We do not use any post-process operation like the CRF [62] used in [8].

Parallel Feature Pyramid (PFP). We first evaluate the proposed PFP module without SeEM. We construct the baseline network by stacking an ASPP module with dilation rates $\{6, 12, 18, 24\}$, as used in [8], at the top of the backbone. The baseline with ResNet50 [24] obtains an mIoU of 70.9%. When constructing PFP without SeEM, we use a convolutional layer with $3 \times 3 \times 512$ filters before fusing the features from the two stages. Table 1 (a)∼(d) show that, by using PFP, 1.8% improvement is obtained for the ResNet50 backbone. When using ResNet101 as the backbone, an mIoU of 75.7% (only 0.8% improvement) is achieved. The limited improvement is caused by the larger semantic gap that exists due to the longer range of feature aggregation for ResNet101 when compared to ResNet50.

Stages	B2	EB2	B3	B3+B2	B3+EB2
mIoU (%)	50.1	62.4	69.4	71.5	73.5

Table 2. Evaluation of different stages of SeENet.

Pixels	1	3	5	10	21	30	40
woBF	50.5	58.3	63.2	69.2	73.9	75.5	76.4
BF-skip	53.2	60.6	64.6	69.9	74.3	75.8	76.7
BAM (ours)	56.1	62.1	66.0	70.8	74.9	76.3	77.2

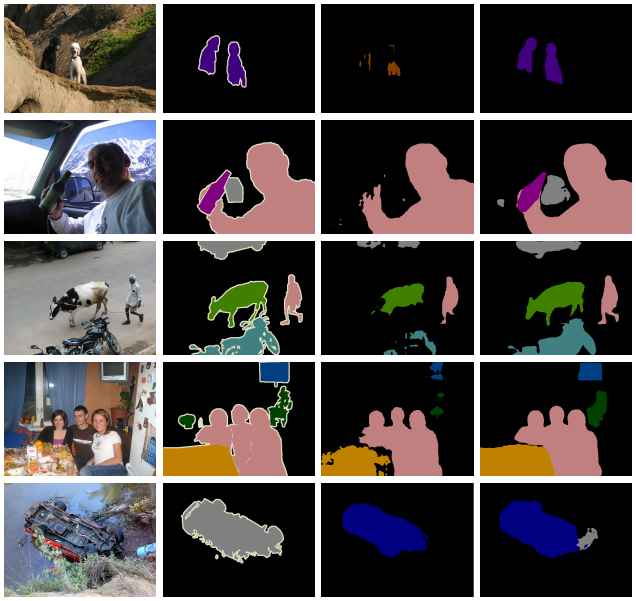
Table 3. Boundary evaluation using trimap measure as [10]. 'woBF' represents without boundary refinement. 'BF-skip' represents boundary refinement with directly skip connection.

Semantic enhancement module. We further adopt the semantic enhancement modules (SeEM, SeEM-FS) to enhance the network. First, we embed three SeEM modules in PFP (*i.e.*, (f)), with the aim of resolving the semantic inconsistency issue. As a result, we obtain an mIoU of 76.9%, which outperforms the model using only a plain pyramid (*i.e.*, (d)) by 1.2% without introducing many parameters. We then evaluate the performance of the proposed SeEM-FS module. If only one SeEM-FS is used on the baseline (c), an mIoU of 75.9% is obtained, which outperforms the baseline by 1.0%. When combining three SeEM-FS modules with the pyramid module, an mIoU of 77.5% is achieved. However, using SeEM-FS causes more parameters to be consumed than when SeEM is used. We achieve a trade-off between parameter consumption and segmentation precision when configuring the SeENet as in Fig. 6. Finally, an mIoU of 77.4% is achieved for (h) in Table 1.

Bridging the semantic gap. We then conduct experiments to demonstrate the effectiveness of SeEM for bridging the semantic gap. Segmentation performance using different stages of SeENet reflects the feature levels of the corresponding features. Using features from different stages of SeENet to predict the segmentation results, we show the results in Table 2. We use the features from Block2 (B2) for prediction and obtain 50.1% mIoU. Using SeEM to enhance the features of B2 (EB2), we obtain 62.4% IoU, which is 12.3% better than that of B2 and 7.0% less than using Block3 (B3) for prediction. It can be found the semantic gap between B2 and B3 can be alleviated by introducing the SeEM. By combining the features of B2 and B3 for prediction, we obtain 71.5% mIoU. By combining with

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU (%)
FCN [36]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
GCRF [45]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [35]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [31]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
ResNet38 [51]	94.4	72.9	94.9	68.8	78.4	90.6	90.0	92.1	40.1	90.4	71.7	89.9	93.7	**91.0**	89.1	71.3	90.7	61.3	87.7	78.1	82.5
PSPNet [60]	91.8	71.9	94.7	71.2	75.8	95.2	89.9	**95.9**	39.3	90.7	71.7	90.5	**94.5**	88.8	**89.6**	72.8	89.6	64.0	85.1	76.3	82.6
AAF [28]	91.3	72.9	90.7	68.2	77.7	95.6	90.7	94.7	40.9	89.5	72.6	**91.6**	94.1	88.3	88.8	67.3	92.9	62.6	85.2	74.0	82.2
TripleNet [5]	**95.6**	70.7	93.3	71.4	78.4	96.2	**92.4**	93.1	**43.0**	89.0	**73.7**	87.4	92.8	89.2	88.5	69.0	92.5	**68.4**	88.1	**80.3**	83.3
EncNet [56]	94.1	69.2	**96.3**	**76.7**	**86.2**	**96.3**	90.7	94.2	38.8	90.7	73.3	90.0	92.5	88.8	87.9	68.7	92.6	59.0	86.4	73.4	82.9
SeENet (ours)	93.7	**73.7**	94.4	67.8	82.4	94.5	90.7	94.1	42.4	**92.5**	72.1	90.8	92.6	88.3	89.4	**76.6**	**92.9**	68.1	**88.5**	77.2	**83.8**
With COCO Pre-training																					
DeepLabv2 [8]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
RefineNet [30]	95.0	73.2	93.5	78.1	84.8	95.6	89.8	94.1	43.7	92.0	77.2	90.8	93.4	88.6	88.1	70.1	92.9	64.3	87.7	78.8	84.2
PSPNet [60]	95.8	72.7	**95.0**	78.9	84.4	94.7	92.0	95.7	43.1	91.0	**80.3**	91.3	96.3	**92.3**	90.1	71.5	**94.4**	66.9	88.8	82.0	85.4
DeepLabv3 [9]	96.4	76.6	92.7	77.8	**87.6**	96.7	90.2	95.4	47.5	93.4	76.3	91.4	97.2	91.0	**92.1**	71.3	90.9	**68.9**	90.8	79.3	85.7
EncNet [56]	95.3	76.9	94.2	**80.2**	85.2	96.5	90.8	96.3	47.9	93.9	80.0	92.4	96.6	90.5	91.5	70.8	93.6	66.5	87.7	80.8	85.9
SeENet (ours)	**97.3**	**81.2**	94.8	77.4	87.5	**97.4**	**92.6**	**96.6**	**48.2**	**94.2**	73.2	**93.7**	**97.2**	91.7	91.5	**72.5**	94.1	66.5	**90.8**	**82.7**	**86.6**

Table 4. Per-class results on the PASCAL VOC 2012 *test* set. The proposed SeENet outperforms most state-of-the-art methods.



(a) Image (b) Ground Truth (c) Baseline (d) Ours

Figure 7. Segmentation results on Pascal VOC 2012 *validation* set. Better boundary in the third and fourth row. The last row shows one failure case.

the enhanced features, we further improve the performance by 2.0%. Therefore, SeEM promotes the feature fusion in our parallel pyramid. Visualization in Fig. 1 (d)∼(f) indicates that SeEM can improve the representation ability of the input features. For the shallow features, as presented in Fig. 1(a), the large semantic gaps for that in (b) have been alleviated by generating the features (d). Fusing features as (d) and (b), which have a similar semantic level, is more robust than fusing (a) and (d).

Boundary attention module. To better evaluate the boundary performance, we use the trimap metric, only calculating the boundary mIoU under different boundary widths, following [8]. As shown in Table 3, if only 1 pixel width is used as the boundary, our BAM obtains 56.1% boundary mIoU, which is much better than when the skip connection method is used. Therefore, our BAM helps maintain better boundary information by discovering the complementary information. This also demonstrates the effectiveness of our attention strategy for robust feature fusion. Table 1 shows that, by using a skip connection (*i.e.*, BF-Skip), a 0.3% mIoU improvement can be achieved. In contrast, the proposed BAM obtains a 0.7% mIoU improvement, outperforming the BF-Skip method. Finally, the proposed SeENet achieves an mIoU of 78.1%.

4.1.2 Performance on Test Set

We first evaluate the effects of crop-size and multi-scale test with ResNet101 as a backbone. When using a smaller crop-size of 320 × 320, we train SeENet with a mini-batch of 10, for 50k iterations on the *training* set and another 50k iterations on the *train_val* set for fine-tuning. When using a larger crop-size of 512 × 512, we train SeENet with a mini-batch of 8, for 70k iterations on the *training* set and another 50k iterations on the *train_val* set for fine-tuning. SeENet achieves an mIoU 80.5% for the model trained with the 320 × 320 input. When using the larger crop-size of 512 × 512, a 1.3% improvement is obtained. By performing inference on multi-scale $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$ inputs, as done in [8, 9, 56, 60], SeENet achieves 83.8% mIoU[1]. With pre-training on COCO dataset following [56], we obtain a better result, with 86.6% mIoU[2] on the *test* set.

We compare our method with other state-of-the-art methods on the VOC2012 *test* set. The IoU for each class and the corresponding mIoU are shown in Table 4. The proposed method shows enhanced overall performance when compared to all other models, including PSPNet [60] and EncNet [56]. In particular, for some difficult classes such as

[1]http://host.robots.ox.ac.uk:8080/anonymous/EN0UWH.html
[2]http://host.robots.ox.ac.uk:8080/anonymous/VJBC6X.html

Methods	road	side	building	wall	fence	pole	T-light	T-sign	vege	terrain	sky	person	rider	car	truck	bus	train	motor	bicycle	mIoU (%)
RefineNet [30]	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
DUC-HDC [46]	98.5	85.5	92.8	58.6	55.5	65.0	73.5	77.9	93.3	72.0	95.2	84.8	68.5	95.4	70.9	78.8	68.7	65.9	73.8	77.6
ResNe-38 [51]	98.5	85.7	93.1	55.5	59.1	67.1	74.8	78.7	93.7	72.6	95.5	86.6	69.2	95.7	64.5	78.8	74.1	69.0	76.7	78.4
DepthSeg [29]	98.5	85.4	92.5	54.4	60.9	60.2	72.3	76.8	93.1	71.6	94.8	85.2	68.9	95.7	70.1	86.5	75.5	68.3	75.5	78.2
AAF [28]	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 [52]	**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 [61]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	80.1
SeENet (ours)	**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**

Table 5. Per-class results on the Cityscapes *test* set. The proposed SeENet obtains an mIoU of 81.2% with training on *fine*-labeled data.

Methods	mIoU (%)
SegNet [2]	46.4
FCN8 [36]	57.0
Dilation8 [54]	65.3
DeepLab [7]	64.6
FC-DenseNet [27]	66.9
SeENet (ours)	**68.4**

Table 6. Results on the CamVid *test* set.

bike and plant, SeENet outperforms others by a large margin. The precise segmentation ability of SeENet is visualized in Fig. 7. It is difficult to tackle the final case because of the context ambiguity (car rarely found in rivers).

4.2. Results on CamVid Dataset

The CamVid dataset [3] is composed of fully segmented videos for urban scene understanding and contains 11 objects classes. We use the same split frames as those in [27]. There are 468 frames (*train_val* set) for training and 233 frames (*test* set) for testing. We use the original image with a resolution 360×480 for training and testing. The mIoU across all 11 classes is used for performance measurement. As shown in Table 6, SeENet achieves an mIoU of 68.4%. Thus, the proposed method is capable of tackling street scenes.

4.3. Results on Cityscapes Dataset

To demonstrate the ability of SeENet for tackling segmentation on high-resolution (2048×1024) complex street scenes, we evaluate the proposed SeENet on the Cityscapes dataset [12]. 5000 of these images have pixel-level annotations (fine labelled in 19 classes). Following the standard settings for Cityscapes, these images are split into 2975 images for the *training* set, 500 images for the *validation* set, and the remaining 1525 images for the *test* set.

We set the crop-size as 768×768 for training and use the original image for testing. We first train our SeENet on the *training* set with a mini-batch size of 8 for $90k$ iterations. We further fine-tune it on the *train_val* set with a smaller learning rate set ($baselr = 0.001$) for another $90k$ iterations. As shown in Table 5, better performance on segmentation for most of the classes is obtained. Finally, our SeENet achieves an mIoU of 81.2%. We visualize some results in Fig. 8. SeENet is able to tackle the complex scale variations and obtain top performance for high-resolution street scene parsing.

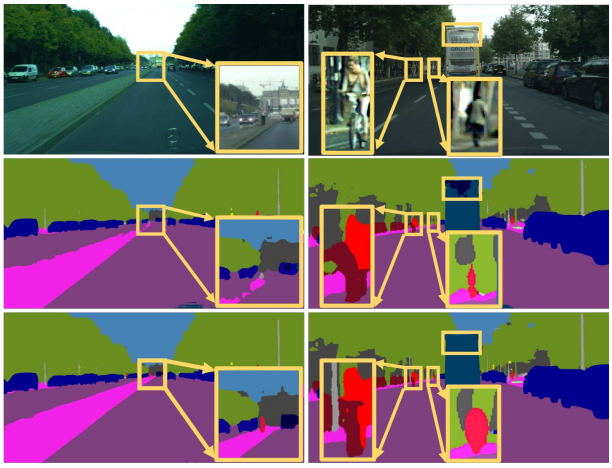


Figure 8. Segmentation predictions on the Cityscapes *test* set. The first row represents the input images, the second shows the baseline results, and the last illustrates our results.

5. Conclusion

Bridging the semantic gap between multi-level features is essential for robust feature aggregation. In this paper, we have proposed a parallel pyramid to aggregate multi-level features in a bottom-up manner. Two strategies have been explored towards bridging the semantic gap and embedded in our parallel pyramid. One strategy is enhancing the representation ability of shallow features to alleviate the semantic inconsistency between multi-level features. Semantic enhancement modules have been designed with this strategy for robust feature fusion. The other strategy is discovering complementary information in very shallow features to enhance deep features. We have designed a boundary attention module with this strategy for boundary refinement. A network, SeENet, with our parallel pyramid has been constructed for semantic segmentation. As a result, SeENet obtains better performance on several benchmark datasets than other state-of-the-art methods, which demonstrates the effectiveness of our method.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China (Grant No. 61632018) and the Beijing Natural Science Foundation under Grant 4182056.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016. 6

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. 1, 2, 8

[3] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 5, 8

[4] Jiale Cao, Yanwei Pang, and Xuelong Li. Exploring multi-branch and high-level semantic networks for improving pedestrian detection. *CoRR*, abs/1804.00872, 2018. 3

[5] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. In *CVPR*, 2019. 1, 7

[6] Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, 2018. 2

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 3, 8

[8] 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. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 2, 3, 5, 6, 7

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 2, 6, 7

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 4, 6

[11] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *ECCV*, 2018. 5

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5, 8

[13] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018. 2

[14] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *ECCV*, 2018. 1

[15] Xingping Dong, Jianbing Shen, Dongming Wu, Kan Guo, Xiaogang Jin, and Fatih Porikli. Quadruplet network with one-shot learning for fast visual object tracking. *IEEE Trans. Image Processing*, 2019. 2

[16] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[17] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 5

[18] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. 2

[19] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *CoRR*, abs/1809.02983, 2018. 1

[20] Jun Fu, Jing Liu, Yuhang Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *CoRR*, abs/1708.04943, 2017. 2

[21] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017. 1

[22] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 6

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[26] Md. Amirul Islam, Shujon Naha, Mrigank Rochan, Neil D. B. Bruce, and Yang Wang. Label refinement network for coarse-to-fine semantic segmentation. *CoRR*, abs/1703.00551, 2017. 2

[27] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPR*, 2017. 2, 8

[28] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X. Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, 2018. 7, 8

[29] Shu Kong and Charless C. Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *CVPR*, 2018. 8

[30] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1, 2, 7, 8

[31] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian D. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 2, 7

[32] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[33] Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, 2018. 3

[34] Shu Liu, Xiaojuan Qi, Jianping Shi, Hong Zhang, and Jiaya Jia. Multi-scale patch aggregation (MPA) for simultaneous detection and segmentation. In *CVPR*, 2016. 2

[35] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 7

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 7, 8

[37] Ping Luo, Guangrun Wang, Liang Lin, and Xiaogang Wang. Deep dual learning for semantic image segmentation. In *ICCV*, 2017. 2

[38] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1, 2

[39] Yanwei Pang, Manli Sun, Xiaoheng Jiang, and Xuelong Li. Convolution in convolution for network in network. *CoRR*, abs/1603.06759, 2016. 2

[40] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters - improve semantic segmentation by global convolutional network. In *CVPR*, 2017. 2, 3, 5

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1, 2

[42] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 2, 3

[43] Hanqing Sun and Yanwei Pang. Glancenets - efficient convolutional neural networks with adaptive hard example mining. *SCIENCE CHINA Information Sciences*, 2018. 3

[44] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 3, 5

[45] Raviteja Vemulapalli, Oncel Tuzel, Ming-Yu Liu, and Rama Chellappa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, 2016. 7

[46] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison W. Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 3, 8

[47] Wenguan Wang, Jianbing Shen, and Haibin Ling. A deep network solution for attention and aesthetics aware photo cropping. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 1

[48] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *IEEE Trans. Image Processing*, 2018. 1

[49] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 1

[50] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *CVPR*, 2018. 3

[51] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *CoRR*, abs/1611.10080, 2016. 7, 8

[52] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018. 2, 8

[53] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018. 1

[54] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015. 3, 8

[55] Fisher Yu, Vladlen Koltun, and Thomas A. Funkhouser. Dilated residual networks. In *CVPR*, 2017. 3

[56] Hang Zhang, Kristin J. Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 3, 5, 6, 7

[57] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017. 1

[58] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018. 2

[59] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *ECCV*, 2018. 1, 2, 3

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

[61] 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. 8

[62] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 6