BOOSTING SEMANTIC SEGMENTATION VIA FEATURE ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic segmentation aims to map each pixel of an image into its corresponding semantic label. Most existing methods either mainly concentrate on high-level features or simple combination of low-level and high-level features from backbone convolutional networks, which may weaken or even ignore the compensation between different levels. To effectively take advantages from both shallow (textural) and deep (semantic) features, this paper proposes a novel plug-and-play module, namely *feature enhancement module* (FEM). The proposed FEM first aligns features from different stages through a learnable filter to extract desired information, and then enhances target features by taking into account the extracted message. Two types of FEM, *i.e.* detail FEM and semantic FEM, can be customized. Concretely, the former type strengthens textural information to protect key but tiny/low-contrast details from suppression/removal, while the other one highlights structural information to boost segmentation performance. By equipping a given backbone network with FEMs, there might contain two information flows, *i.e.* detail flow and semantic flow. Extensive experiments on the Cityscapes, PASCAL Context, and ADE20K datasets are conducted to validate the effectiveness of our design, and reveal its superiority over other state-of-the-art alternatives.

1 INTRODUCTION

Semantic segmentation aims to assign a semantic label to each pixel of a given image, which is in nature a classification task. As a fundamental component in a variety of practical tasks, such as image editing (Aksoy et al., 2018), autonomous driving (Teichmann et al., 2018), robot sensing (Hua et al., 2019), this problem has been drawing much attention from computer vision and machine learning communities with great progress made over past years.

Recent deep learning based schemes, contrary to traditional methods that rely on hand-crafted features, have shown their clear advances thanks to the strong capability of feature learning. For semantic segmentation, features extracted from deeper layers in a given network typically reflect higher-level/semantic information, and finally map to semantic maps. However, merely concentrating on high-level features often gradually loses some important clues from those generated by shallower layers (Long et al., 2015). To mitigate this issue, an intuitive and commonly-adopted manner is to directly aggregate features from different layers (Ronneberger et al., 2015). Although improving performance to some extent, such a simple operation may weaken the compensation between features at different levels.

Let us here consider a more general problem, *i.e.*, classification, the challenge of which mainly comes from the small inter-class distance and large intra-class variance of samples. Although semantic segmentation has its own characteristics compared with *e.g.* image classification, it suffers from the same difficulty. Specifically, the following summarizes issues regarding the segmentation accuracy into two aspects: 1) **Key but low-contrast/tiny details missing**. This issue is often caused by either indistinguishable appearances of spatially-nearby regions belonging to different classes, or over-large receptive fields, both of which require detail enhancement in the feature domain to enlarge the inter-class distance, and thus avoiding suppression or removal. Please see the "motorcycle" and "pole" marked by the white boxes in Fig. 1; and 2) **Diverse patterns of same class misunderstanding.** Objects of one certain class may have different and complex looks. Higher intra-class variance makes the prediction considerably more difficult due to the multiple-to-one nature of map-

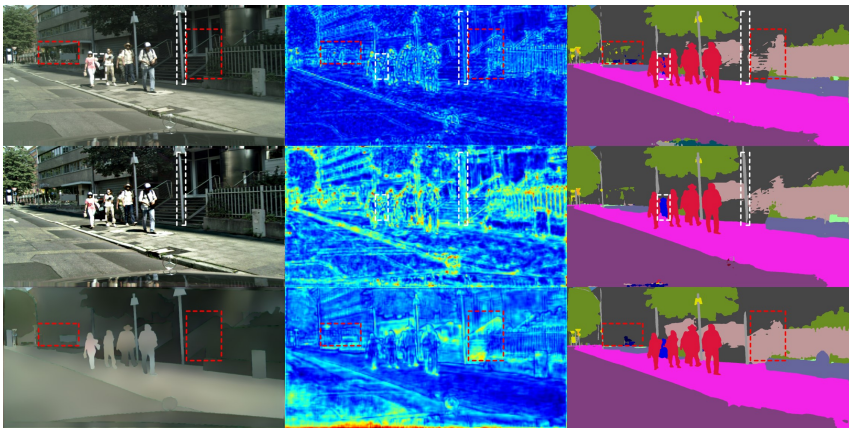

Figure 1: The top row shows a sample image (left column), one of intermediate features (middle column) and the final prediction (right column), while the middle and bottom rows provide results with detail enhancement and structure (or semantic) enhancement, respectively.

ping. To mitigate this issue, restricting trivial features and highlighting structural ones (semantic enhancement) is a possible solution. Please see the "building" and "fence" marked by the red boxes in Fig. 1.

To alleviate the former issue, a number of methods attempt to connect deeper features with shallower ones for maintaining better details. As a representative, Gated-SCNN (Takikawa et al., 2019) employs deeper feature activations to gate the shallower ones for refining feature boundaries. The result of this method highly depends on the learned deep features. In addition, SFNet (Li et al., 2020b) adopts semantic flow to align features from different layers. Though this method shows its effectiveness, the performance is limited by the quality of semantic flow, which is not explicitly supervised. Most recently, STLNet (Zhu et al., 2021) uses statistic texture information to enhance deep features with improved performance. As for the latter issue, a variety of schemes concentrate on modeling the global context. For example, PSPNet (Zhao et al., 2017) develops a pyramid pooling module (PPM) to aggregate features with various sizes of pooling kernels. The attention models (Zhao et al., 2018; Fu et al., 2019; Yuan et al., 2018) acquire the global context for each pixel by modeling the relationship between the query and key pixels. Despite these methods have achieved impressive results, they need to generate immense attention maps, which are computationally expensive. More recently, several works (Yuan et al., 2020; Shen et al., 2020; Cheng et al., 2021) further seek the relationship between pixel and category to obtain more robust global context. In the literature, existing methods barely consider the two aspects simultaneously. Based on the above analysis, *investigating more principled ways to effectively utilizing features is crucial to further boost the performance of semantic segmentation.*

Motivation & Contribution

In the field of low-level vision, the structure preserving image smoothing (Liu et al., 2021; Fan et al., 2018) and image sharpening (Yin et al., 2019) can be viewed as semantic and detail enhancement in the image domain, respectively. A natural question arises: *Can such operations be applied to intermediate deep features?* This paper answers the question by designing a novel plug-and-play module, namely *feature enhancement module* (FEM). Figure 1 illustrates the analogy between image enhancement and feature enhancement. Generally, our proposed FEM first aligns features from different stages through a learnable filter to extract desired information, and then enhances target features by taking in the extracted message. Based on this general design, detail FEM (FEM-D) and semantic FEM (FEM-S) are derived by learning desired filters to respectively protect key but tiny/low-contrast details and highlight structural information, which can be flexibly plugged into different backbones. According to different enhancement purposes, in a certain network equipped with FEMs, multiple FEM-Ds and FEM-Ss constitute *detail flow* (DF) and *semantic flow* (SF), respectively. Extensive experiments together with ablation studies on the Cityscapes, Pascal Context,

and ADE20K benchmarks are conducted to demonstrate the efficacy of our design, and reveal its superiority over state-of-the-art competitors.

2 Related Work

This section will briefly review classic and contemporary techniques that are closely related to ours.

Semantic Segmentation. Fully convolutional networks (FCNs) (Long et al., 2015) are the first attempt to replace the fully connected layer by the convolutional layer for the task of semantic segmentation. Since then, its follow-ups, *i.e.* FCN-based methods, have made great progress via better exploiting contextual information. As a representative technical line, the DeepLab series (Chen et al., 2017a;b) adopt astrous spatial pyramid pooling (ASPP) to enhance the pixel representation with multi-scale contextual aggregation, which consists of parallel dilated convolutions with different dilated rates. Moreover, PSPNet (Zhao et al., 2017) makes use of the pyramid pooling module (PPM) to capture multi-scale context. Though these methods achieve certain gain, the contextual range dragged into consideration is still local, and thus limiting the performance. To capture context over the whole image, some methods (Yuan et al., 2018; Fu et al., 2019; Zhang et al., 2019c) have been proposed through modeling pixel-to-pixel affinities. However, the above methods suffer from heavy computational overhead on generating attention maps. To reduce the computation complexity of self-attention, CCNet (Huang et al., 2019) leverages two criss-cross attention modules. Instead of modeling the relationship between pixels, ACFNet (Zhang et al., 2019b) and OCRNet (Yuan et al., 2020) obtain the global context by modeling the pixel-class and pixel-region relationships respectively. Different from these methods, we enhance features by combining the complementary information of cross-level features with a simple module.

Feature Fusion. In addition to global context pursuit, multi-level feature fusion is frequently used in semantic segmentation to refine deep features with shallow ones. To obtain more detailed features, FCN (Long et al., 2015) gradually combines the features of the last layer with the previous features. The UNet (Ronneberger et al., 2015) adds skip connections between the encoder and decoder to keep the detail information of low-level features. Despite these strategies show positive effects, they ignore the representation gap between different feature layers. BiSeNet (Yu et al., 2018) constructs an extra spatial path to refine features coming from the context path for restoring details. Furthermore, in order to remove structure redundancy in BiSeNet (Yu et al., 2018), STDCNet (Fan et al., 2021) integrates the learning of spatial information into low-level layers in a single-stream manner. More recently, GFFNet (Li et al., 2020c) uses gates to selectively fuse features from multiple levels. Li *et al.* (Li et al., 2020b) introduced a flow alignment module to connect the adjacent features via the learned semantic flows. Though it shows noticeable merits, the performance greatly depends on the quality of generated semantic flows, which is not always guaranteed due to its unsupervised nature. Unlike these methods, we introduce a simple module, named feature enhancement module, to effectively eliminate the representation gap between multi-level features.

3 Methodology

For semantic segmentation, there are two main logical components, *i.e.*, feature extractor and classifier. The focus of this paper is on the feature/representation part via proposing a general strategy.

3.1 Problem Analysis

The goal of semantic segmentation is to map an input image $I \in \mathbb{R}^{H\times W\times C}$ to a semantic map $Y \in \mathbb{R}^{H\times W\times 1}$ with the same spatial resolution $H\times W$. To accurately predict the semantic map from the input, learning-based algorithms need to seek discriminative and robust representations, which are then fed into the classifier head. Suppose we have a set of features $F \in \mathbb{R}^{H^l\times W^l\times C^l}$ extracted by a given deep network, where the spatial size $H^l \times W^l$ and channel number C^l could vary at different layers. In addition, there exists a corresponding ideal feature group $\tilde{F}$ expected by the classifier. Formally, we can always have $\tilde{F} = F + E$, where E stands for the residual between the ideal and learned features. The problem can be resolved if we could retrieve the residual. However, hardly the target $\tilde{F}$ is available in practice. As discussed previously, the degraded performance usually comes from insufficient abilities in handling *key but low-contrast/tiny details missing* (small inter-class

distance) and *diverse patterns of same class misunderstanding* (large intra-class variance). Hence, we alternatively consider how to enhance such abilities for boosting the performance.

Considering an image contains rich information like the first one shown in Fig. 1, to clearly distinguish objects, one may need to highlight boundaries between objects and suppress complex textures within each individual. This goal can be accomplished by enhancement techniques in image processing, say detail enhancement and structure-preserving smoothing, both of which are derived from the concept of filtering. Inspired by the above, we can naturally design similar functions for learned features. In what follows, we will introduce a general module namely FEM to effectively strengthen desired information in deep features against the aforementioned challenges, and show how to flexibly plug it into different backbones.

3.2 FEM: Feature Enhancement Module

Due to the complementarity between cross-level features, our principle is to mine important information from features of one level (auxiliary F^a), and supply it to those of another level (target F^t). Concretely, given F^t and F^a with the same channel number, we first feed both features respectively into learnable information extractors (filters) to capture the desired information. Having the extracted information from two feature groups, we resize the spatial size of the auxiliary features to the same size of the target ones. After some refinement by convolutional and residual blocks, the refined message is added to the auxiliary/target features for the sake of enhancement. The above procedure can be simply written as:

$$\tilde{F}^t = \text{FEM}(F^t, F^a) = \Theta(F + \text{Oper}(F^t, F^a)), \tag{1}$$

where F can be F^t or F^a for different purposes, Θ is a 3×3 convolution, and the function $\text{Oper}(F^t, F^a)$ contains size adjustment, information extraction, and refinement.

As shown in Fig. 2, there are two types of FEM with respect to detail FEM (FEM-D) and semantic FEM (FEM-S). The FEM-D and FEM-S are used for enhancing detail information and semantic information respectively of the target feature F^t. To enhance the desired information, FEM-D and FEM-S first use a detail extractor and a structure extractor to extract the corresponding information respectively. Concretely, given a feature map F, the detail/structure extractor first extracts the residual information[1] through a convolution layer, and then it minus the residual information to form the desired feature map. In practice, the convolution layer is initialized by a mean blur kernel. For desired feature map $\hat{F}$, the process of detail/structure extractor can be formulated as:

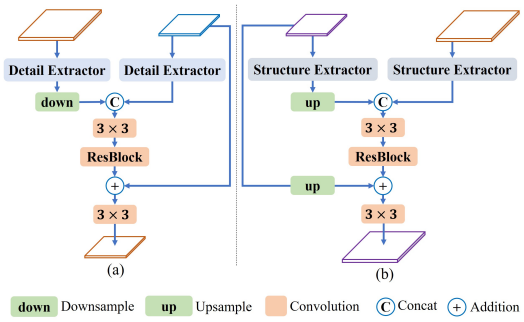

Figure 2: The architectures of FEM-D (left) and FEM-S (right) are depicted, respectively. The operations including batch normalization and ReLU activation are omitted for simplification.

$$\hat{F} = F - K \otimes F, \tag{2}$$

where $F \in \mathbb{R}^{H \times W \times C}$, and K is the convolution kernel. To enhance the details of target feature map F^t, we downsample the detailed feature map $\hat{F}^a$ to the same size as $\hat{F}^t$ via the standard bilinear interpolation. Then the downsampled $\hat{F}^a$ is concatenated with the $\hat{F}^t$, and the concatenated features are further fed into a 3×3 convolution layer and a resBlock module to align $\hat{F}^t$ to $\hat{F}^a$. After that, we add the original target feature map F^t to keep the original information. Finally, we leverage a 3×3 convolution layer to refine the detail enhanced feature map $\tilde{F}^t$. Similarly, for enhancing semantics of low-level feature map, we use the same strategy with only a slight change, such as the downsample operation is changed to upsample operation and the added original target feature map is changed to added upsampled auxiliary feature map. Mathematically, the aforementioned steps for enhancing details and semantics can be formulated respectively as:

$$\tilde{F}^t = \Theta(F^t + \Phi(\text{cat}(\hat{F}^t, \text{down}(\hat{F}^a)))), \tag{3}$$

[1]The difference between original feature map and the desired feature map.

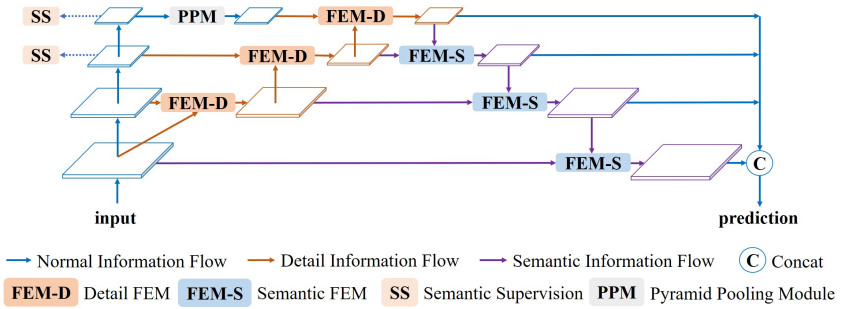

Figure 3: Our architecture contains a backbone network (*e.g.* ResNet (He et al., 2016)), a pyramid pooling module (PPM) (Zhao et al., 2017), and two information flows, *i.e.* detail flow and semantic flow. Both of the two flows are composed of FEMs with different enhancement purposes.

$$\tilde{F}^t = \Theta(\text{up}(F^a) + \Phi(\text{cat}(\hat{F}^t, \text{up}(\hat{F}^a)))), \tag{4}$$

where down$(\cdot)$ is the downsample operation, up$(\cdot)$ is the upsample operation, cat$(\cdot)$ is the concatenation operation, Φ indicates a 3×3 convolution layer and a resBlock module, and Θ is the last 3×3 convolution layer.

On the whole, our FEM module is a general module, which can not only enhance detail information but can also enhance semantic information. In both cases, we design a general information extractor to extract detail or structure information. For equation 3, we use the aligned features with rich detail information extracted from the auxiliary feature map F^a and the target feature map F^t to enhance the target feature map F^t. While in equation 4, the aligned features with abundant structure information extracted from F^a and F^t are used. Moreover, we further enhance the target feature map F^t by adding the upsampled auxiliary feature map F^a.

3.3 An Illustrative Network Architecture

To verify our primary claims in this paper, we provide an illustrative network architecture equipped with our two special designed FEMs in Fig. 3. As the figure shows, the input is first fed into a pre-trained backbone network (*e.g.* ResNet (He et al., 2016)) from ImageNet (Russakovsky et al., 2015) for feature extraction. The backbone network consists of four stages, each of which contains a convolution layer with stride 2 to downsample the feature map in order to improve computational efficiency and obtain a larger receptive fields. Following SFNet (Li et al., 2020b), we add a Pyramid Pooling Module (PPM) (Zhao et al., 2017) at the last stage of backbone network to aggregate rich contextual information. Since our FEM module is general and can enhance details and semantics, we design two information flows, named detail information flow and semantic information flow, to enhance details and semantics of the extracted features based on the corresponding FEMs respectively. Thus, the detail information from the lowest features can be transmitted to the highest features through the detail information flow. Similarly, the semantic information can be transmitted from the highest features to the lowest features through the semantic information flow. After the two information flows, features from each stage are concatenated for the final prediction.

Because the backbone network is trained for image classification, the features may not suit for the semantic segmentation paradigm. To alleviate the representation gap of features between these two tasks, we add auxiliary semantic supervisions at these features. Instead of these auxiliary loss functions, we use the principle loss function to supervise the output of the whole network. Finally, the total loss can be formulated as:

$$\mathcal{L} = \mathcal{L}_p + \lambda \sum_{i=1}^{S} \mathcal{L}_{s_i}, \tag{5}$$

where $\mathcal{L}_p$ is the principle loss of the final output, $\mathcal{L}_{s_i}$ is the auxiliary semantic loss, S is the number of stages being supervised. Particularly, all of the semantic loss terms adopt cross-entropy based on online hard example mining (Shrivastava et al., 2016). The parameter λ is used to balance the principle and the auxiliary loss terms, which is simply set to 1 for all reported experiments in this

Table 1: Ablation study on the Cityscapes validation set. † means the strong baseline.

Method	DF	SF	SS(4)	SS(2)	mIoU(%)	Δ a
baseline	-	-	-	-	73.2	-
baseline†	-	-	-	-	77.3	-
	✓				78.8	1.5↑
		✓			78.9	1.6↑
	✓	✓			79.6	2.3↑
	✓	✓	✓		80.0	2.7↑
	✓	✓		✓	80.1	2.8↑

(a) On different configurations with ResNet18 as backbone. SS means semantic supervision.

Method	mIoU(%)	Δ a
DFNet1(Li et al., 2019)	70.6	-
DFNet1†(Li et al., 2019)	72.8	-
w/ FEM	74.4	1.6↑
DFNet2(Li et al., 2019)	74.9	-
DFNet2†(Li et al., 2019)	75.8	-
w/ FEM	77.3	1.5↑
ResNet50(He et al., 2016)	75.4	-
ResNet50†(He et al., 2016)	79.3	-
w/ FEM	81.7	2.4↑
ResNet101(He et al., 2016)	78.3	-
ResNet101†(He et al., 2016)	80.5	-
w/ FEM	82.0	1.5↑

(b) On various backbones.

paper to largely focus on the effectiveness of our module design. Please notice that the performance could be further improved by fine-tuning λ.

4 Experimental Validation

4.1 Implementation Details

All our networks are implemented in PyTorch (Paszke et al., 2017). The standard SGD (Krizhevsky et al., 2012) is used for training networks, with momentum of 0.9 and weight decay of $1e^{-4}$. As a common practice, we apply the "poly" learning rate policy to adjust the learning rate by multiplying $(1 - \frac{epoch}{max\_epoch})^{1.0}$ during training. Synchronized batch normalization (Zhang et al., 2018) is employed across multiple GPUs to stabilize the training. For data augmentation, we apply random color jittering, random gaussian blurring, random horizontal flipping, random cropping and random resizing with scale range of $[0.5, 2.0]$. Specifically for Cityscapes, we set the initial learning rate as 0.01, crop size as 1024×1024, batch size as 8 and training epochs as 300. For Pascal Context and ADE20K, the images are cropped into 512×512 and the network is trained for 120 epochs with batch size 16. The initial learning rates are set to $1e^{-3}$ and $1e^{-2}$, respectively.

4.2 Experiments on Cityscapes Dataset

The Cityscapes dataset contains 5000 finely annotated images, which is further divided into 2975, 500, and 1525 images for training, validation, and testing, respectively. Each image in this dataset is of 2048×1024 resolution and contains 19 classes for semantic segmentation. It is worth noting that we only use the fine data in all our experiments. Following GFFNet (Li et al., 2020b), the uniformly sampling strategy is used for this dataset.

Strong Baselines. For all involved backbones, we first compress all the features from four layers to the same channel depth through a 3×3 convolution layer. Then, we use bilinear interpolation to upsample all-stage features into the same spatial resolution as the first stage. Finally, we add all of these features to construct the baseline. Furthermore, to validate our method more convincingly, we introduce extra modules to the baseline to construct the strong baselines. With the ResNet (He et al., 2016) backbone, we add the pyramid pooling module (PPM) (Zhao et al., 2017) to the last stage and modify the last two stages with dilated convolution but keep the strides unchanged. As for other backbones, we only add the PPM module to the last stage.

Ablation Study. To verify the effectiveness of the proposed modules, we first conduct an ablation study on the Cityscapes validation set. As shown in Table 1 (a), the original ResNet18 with addition as the aggregation architecture only obtains 73.2% mIoU. To make it more challenging, we modify the baseline with dilated convolution and pyramid pooling module to construct the strong baseline, which achieves 77.3% mIoU. By applying the detail flow (DF) to the strong baseline, we obtain an improvement of 1.5%, while, by adopting the semantic flow (SF), the improvement of mIoU is 1.6%. From these results, we can see that our detail flow (DF) and semantic flow (SF) can both bring great benefits to the semantic segmentation. Consequently, we

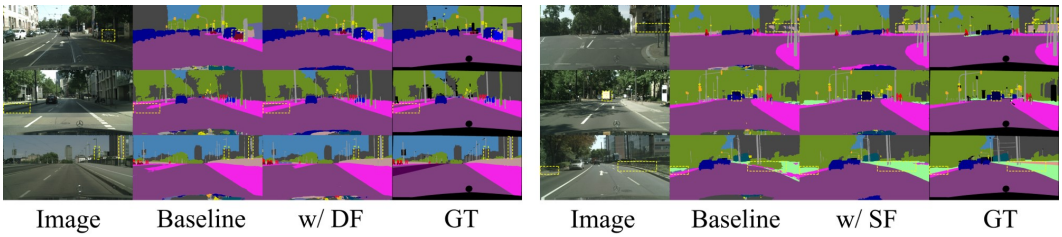

Figure 4: Left: comparison of segmentation results between baseline† and w/ DF on the Cityscapes validation set. Right: comparison of segmentation results between baseline† and w/ SF on the Cityscapes validation set. Best view in color and zoom in.

employ both the two flows to the strong baseline, which further boost the performance in mIoU by 2.3%, confirming the proposed two flows are complementary to each other. After that, we study the impact of auxiliary losses added to the backbone features. By adding the semantic supervision on all four stages, the improvement of mIoU is 0.4%. While with executing the semantic supervision on the last two stages, the mIoU improves 0.5%, reaching 80.1%. Instead of the segmentation accuracy, we also report the computation cost and parameters of the proposed modules in Table 2. As the table shows, both the DF and SF have only 8.9M parameters.

In order to better reveal the advantages of our design visually, we have provide several generated segmentation maps in Fig. 4 and 5 from the Cityscapes validation set. From the left side of Fig. 4, we can observe that after adding the detail flow (DF), the model can generate accurate results at the low contrast areas, *e.g.* the "sidewalk" and the "pole" marked by the yellow boxes. Furthermore, DF can make the objects with low contrast be easier recognized, *e.g.* the "motorcycle" in row one. Furthermore, from the right side of Fig. 4, we can see that SF can produce more consistent segmentation results compared with the baseline. As shown in Fig. 5, having both the DF and SF equipped, the model improves prediction quality at both low contrast areas and objects with high variance appearance.

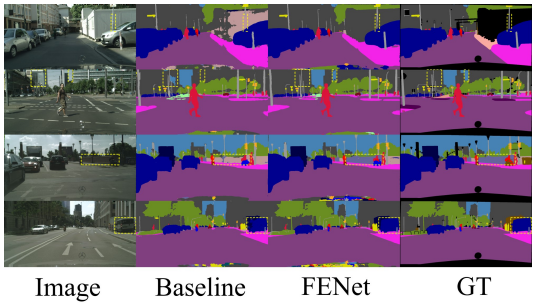

Figure 5: Comparison of segmentation results between baseline† and FENet on the Cityscapes validation set. Best view in color and zoom in.

Robustness on different backbones. To verify the generalization ability of the proposed method, we further carry out experiments on different backbones, including both light-weight and heavy-weight backbones. For light-weight backbones, we choose the DFNet1 (Li et al., 2019) and DFNet2 (Li et al., 2019) as candidates. While for the heavy-weight backbones, the ResNet50 (He et al., 2016) and ResNet101 (He et al., 2016) are involved. To be noted, all the backbones are pretrained on the ImageNet (Russakovsky et al., 2015) dataset. As shown in Table 1(b), our method can greatly improve the performance on all the backbones.

Table 2: Comparison in computation cost with input size of 1024×1024, where ResNet101 is used as backbone. † means the strong baseline.

Method	**FLOPs(G)**	**Params(M)**
baseline	239.2	44.2
baseline†	239.5	46.6
DF	63.5	8.9
SF	254.0	8.9

Visualization of features. To explore the internal reasons why our modules work, we offer the visualized features in Fig. 6. After the detail flow (DF), the features are enhanced with more details, especially on the boundaries. This module can solve the challenge of small inter-class distance and make the objects easier to be identified. While for the semantic flow (SF), the features are enhanced

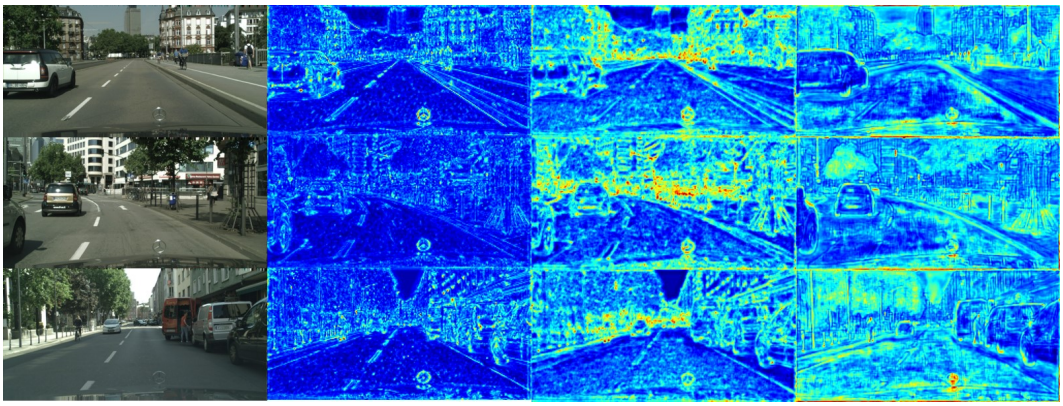

Figure 6: Visualization of features maps. From left to right: input images, original features, features after executing DF, and features after executing SF, respectively.

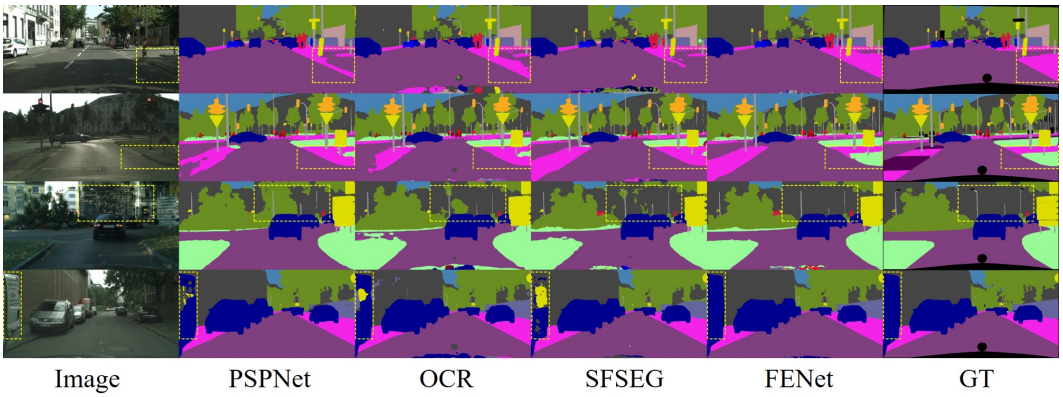

Figure 7: Visual comparison with state-of-the-art methods.

to be more semantically consistent on the interior of the objects. This module is able to effectively solve the challenge of large intra-class variance and remove noise.

Comparisons with other state-of-the-arts. To compare our FENet with state-of-the-art methods, we conduct experiments on the Cityscapes validation and testing set respectively. As for the Cityscapes validation set, we only use the train-fine data for training and use the single scale image for testing without any trick. As Table 3 (a) shows, our FENet with ResNet18 (He et al., 2016) as backbone achieves 80.1% mIoU with 16.8M parameters. It can be observed that the parameters are significant less than other models, but with a promising accuracy. What is more, with the ResNet101 (He et al., 2016) as backbone, our FENet achieves state-of-the-art performance with 82.0% mIoU. As state-of-the-art methods usually train on the mixed data set of training and validation, and perform multi-scale, horizontal flip and dense crop inference to acquire better results on the Cityscapes test server. To make comparisons fair, we also adopt these strategies to report our result on the Cityscapes testing set. As Table 3 (b) lists, our FENet with backbone of ResNet18(He et al., 2016) achieves a compelling result of 81.0% mIoU. In addition, our method achieves state-of-the-art result with ResNet101 (He et al., 2016) backbone. Figure 7 shows several visual comparisons between FENet and state-of-the-arts, as can be seen from which, FENet can generate more consistent result in high variance appearance of objects and generate accurate prediction at low contrast areas.

4.3 Results on Other Datasets

ADE20K Dataset. ADE20K is a challenging semantic segmentation benchmark, which contains 20K/2K images for training and validation. Images in this dataset are densely annotated with 150 classes and from different scenes with various scares. Following the previous work (Li et al., 2020b), we train the network for 120 epochs with batch size 16, crop size 512 and initial learning rate

Table 3: Comparisons on the Cityscapes validation and testing set. The ResNet101 (He et al., 2016) is used as backbone for all models unless explicitly mentioned. $\flat$ means using sliding window crop with horizontal flip for testing.

Method	mIoU(%)	Params(M)
ACFNet(Zhang et al., 2019a)	78.0	-
Gated-SCNN(Takikawa et al., 2019)	80.8	-
CCNet(Huang et al., 2019)	79.8	68.8
OCR(Yuan et al., 2020)	79.6	55.5
GFFNet(Li et al., 2020c)	81.2	70.5
SFNet$^{\flat}$(Li et al., 2020b)	79.8	50.3
FENet(ResNet18)	**80.1**	16.8
FENet(ResNet101)	**82.0**	66.0

(a) Results on Cityscapes validation set.

Method	mIoU(%)
PSPNet(Zhao et al., 2017)	78.4
PSANe(Zhao et al., 2018)	80.1
ACFNet(Zhang et al., 2019a)	81.8
CCNet(Huang et al., 2019)	81.4
DANet(Fu et al., 2019)	81.5
OCR(Yuan et al., 2020)	81.8
SpyGR(Li et al., 2020a)	81.6
GFFNet(Li et al., 2020c)	82.3
SFNet(Li et al., 2020b)	81.8
STLNet(Zhu et al., 2021)	82.3
FENet(ResNet18)	**81.0**
FENet(ResNet101)	**82.3**

(b) Results on Cityscapes testing set.

Table 4: Comparisons on the ADE20K validation set and Pascal Context validation set.

Method	Backbone	mIoU(%)
PSPNet(Zhao et al., 2017)	ResNet101	43.29
EncNet(Zhang et al., 2018)	ResNet101	44.65
CFNet(Zhang et al., 2019c)	ResNet101	44.89
CCNet(Huang et al., 2019)	ResNet101	45.22
OCR(Yuan et al., 2020)	ResNet101	45.28
GFFNet(Li et al., 2020c)	ResNet101	45.33
SFNet(Li et al., 2020b)	ResNet101	44.67
STLNet(Zhu et al., 2021)	ResNet101	**46.48**
FENet	ResNet101	45.42

(a) Results on ADE20K validation set.

Method	Backbone	mIoU(%)
RefineNet(Lin et al., 2017)	ResNet152	47.3
PSPNet(Zhao et al., 2017)	ResNet101	47.8
EncNetZhang et al. (2018)	ResNet101	51.7
DANet(Fu et al., 2019)	ResNet101	52.6
DMNet(He et al., 2019)	ResNet101	54.4
GFFNet(Li et al., 2020c)	ResNet101	54.2
STLNet(Zhu et al., 2021)	ResNet101	**55.8**
FENet	ResNet101	53.7

(b) Results on Pascal Context validation set.

$1e^{-2}$. Finally, we perform multi-scale testing with horizontal flip and dense crop operation on the validation dataset. As shown in Table 4(a), our method achieves 45.42% mIoU. To be noted, although the stride of our method is 32 and many images are with a small resolution in this dataset, our method still gets a compelling result.

Pascal Context Dataset. Extending from Pascal Voc 2010, Pascal Context provides more detailed segmentation annotation for 59 classes. There are 4998 images for training and 5105 images for validation. Following the previous work (Li et al., 2020b), we train the network for 120 epochs with batch size 16, crop size 512 and initial learning rate $1e^{-3}$. Finally, we perform multi-scale testing and horizontal flip operation on the validation dataset. As reported in Table 4 (b), our approach reaches 53.7% mIoU. To be noted, the stride of our method is 32 and images in this dataset are with a small resolution. We again emphasize that the only hyper-parameter λ in the objective function is uniformly set to 1 for all the experiments, which can be carefully tuned for different datasets to seek higher performance.

5 Conclusion

This paper has proposed a general module named Feature-Enhancement Module (FEM) to effectively shrink the representation gap between cross-level features. The novel module can not only enhance features with detail information, but can also enhance semantic information with little change. Based on this, we designed two information flows, one is for transmitting detail information from low-level features, the other is for transmitting semantic information from high-level features. Extensive experiments on challenging benchmarks have shown the effectiveness of the proposed method. The implementation is quite simple and our code will be made publicly available.

REFERENCES

Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

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 transactions on pattern analysis and machine intelligence*, 40(4): 834–848, 2017a.

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

Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint arXiv:2107.06278*, 2021.

Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9716–9725, 2021.

Qingnan Fan, Jiaolong Yang, David Wipf, Baoquan Chen, and Xin Tong. Image smoothing via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, 2019.

Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3562–3572, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Minjie Hua, Yibing Nan, and Shiguo Lian. Small obstacle avoidance based on rgb-d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 603–612, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8950–8959, 2020a.

Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision*, pp. 775–793. Springer, 2020b.

Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11418–11425, 2020c.

Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9145–9153, 2019.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1925–1934, 2017.

Wei Liu, Pingping Zhang, Yinjie Lei, Xiaolin Huang, Jie Yang, and Michael Kwok-Po Ng. A generalized framework for edge-preserving and structure-preserving image smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Dingguo Shen, Yuanfeng Ji, Ping Li, Yi Wang, and Di Lin. Ranet: Region attention network for semantic segmentation. *Advances in Neural Information Processing Systems*, 2020.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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)*, pp. 1013–1020. IEEE, 2018.

Hui Yin, Yuanhao Gong, and Guoping Qiu. Side window filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8758–8766, 2019.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018.

Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 173–190. Springer, 2020.

Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019a.

Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6798–6807, 2019b.

Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7151–7160, 2018.

Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 548–557, 2019c.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 267–283, 2018.

Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12537–12546, 2021.