Consistent-Separable Feature Representation for Semantic Segmentation

Xingjian He,[1,2] Jing Liu,[1,2*] Jun Fu,[2] Xinxin Zhu,[2] Jinqiao Wang,[2] Hanqing Lu[1,2]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
{xingjian.he, jliu, jun.fu, xinxin.zhu, jqwang, luhq}@nlpr.ia.ac.cn

Abstract

Cross-entropy loss combined with softmax is one of the most commonly used supervision components in most existing segmentation methods. The softmax loss is typically good at optimizing the inter-class difference, but not good at reducing the intra-class variation, which can be suboptimal for semantic segmentation task. In this paper, we propose a Consistent-Separable Feature Representation Network to model the Consistent-Separable (C-S) features, which are intra-class consistent and inter-class separable, improving the discriminative power of the deep features. Specifically, we develop a Consistent-Separable Feature Learning Module to obtain C-S features through a new loss, called Class-Aware Consistency loss. This loss function is proposed to force the deep features to be consistent among the same class and apart between different classes. Moreover, we design an Adaptive feature Aggregation Module to fuse the C-S features and original features from backbone for the better semantic prediction. We show that compared with various baselines, the proposed method brings consistent performance improvement. Our proposed approach achieves state-of-the-art performance on Cityscapes (82.6% mIoU in test set), ADE20K (46.65% mIoU in validation set), COCO Stuff (41.3% mIoU in validation set) and PASCAL Context (55.9% mIoU in test set).

Introduction

Semantic segmentation is one of the most challenging and fundamental problems in computer vision, which aims to assign per-pixel class label for a given image. It has been applied to various challenging fields, such as autonomous driving, image editing and human parsing, etc. Benefiting from the Fully Convolution Networks (FCN) (Long, Shelhamer, and Darrell 2015), recent approaches achieve promising performance. FCN-based methods usually adopt softmax loss as an optimization function to learn discriminative features for pixel-level semantic segmentation. However, the softmax loss typically does well in enlarging the inter-class difference (i.e. separating different classes), but not good at shrinking the intra-class diversity (Wang et al. 2018a; Liu et al. 2017a). This may cause inconsistent predictions of the same category. From the view of feature discriminative ability, the intra-class consistency and the inter-class separability are equally important for classification.

To enhance the intra-class consistency, some works adopt feature aggregation with self-attention mechanism, which implicitly alleviates the intra-class diversity, enhancing the discriminative ability of feature representations. For example, Nonlocal (Wang et al. 2018b) introduces a self-attention mechanism to capture pair-wise relationships and utilizes the relationships to guide feature fusion, so that similar features tend to be more similar, which enhances the intra-class consistency implicitly. CPNet (Yu et al. 2020b) models the intra-class dependency through aggregating the pixels belonging to the same category, shrinking the intra-class variance and achieving good segmentation performance. ACFNet (Zhang et al. 2019a) builds class-aware context features through adaptively combining different context according to the category of each pixel, reducing the intra-class variation. These previous methods adopt an implicit feature learning scheme to enhance the similar or intra-class features by themselves along a bottom-up direction.

In this paper, we propose a Consistent-Separable Feature Representation Network (CSFRN) to obtain discriminative features for per-pixel semantic prediction. Different from the previous works based on feature aggregation, we adopt a top-down supervised scheme to explicitly learn the intra-class consistent and inter-class separable features for the per-pixel semantic prediction. Specifically, we design a Consistent-Separable Feature Learning Module (CSFLM) on the top of the dilated FCN. The process of this module could be divided into three steps. First, we generate the feature embedding for each pixel from the output feature of the dilated FCN. And we build class-wise semantic centers by using the feature embeddings and ground truth information. Second, we design a new loss function, namely Class-Aware Consistency (CAC) loss, to simultaneously minimize the distances from the feature embedding of each pixel to its corresponding class semantic center, and maximize the distances from the feature embedding of each pixel to other class semantic centers. Considering the degree of intra-class compactness of different categories should be different, in this loss function, we introduce a class-aware margin mechanism to adaptively adjust the degree for different categories. With the supervision of CAC loss, we could obtain the Consistent-Separable (C-S) features. Third, we fuse the

*Corresponding Author

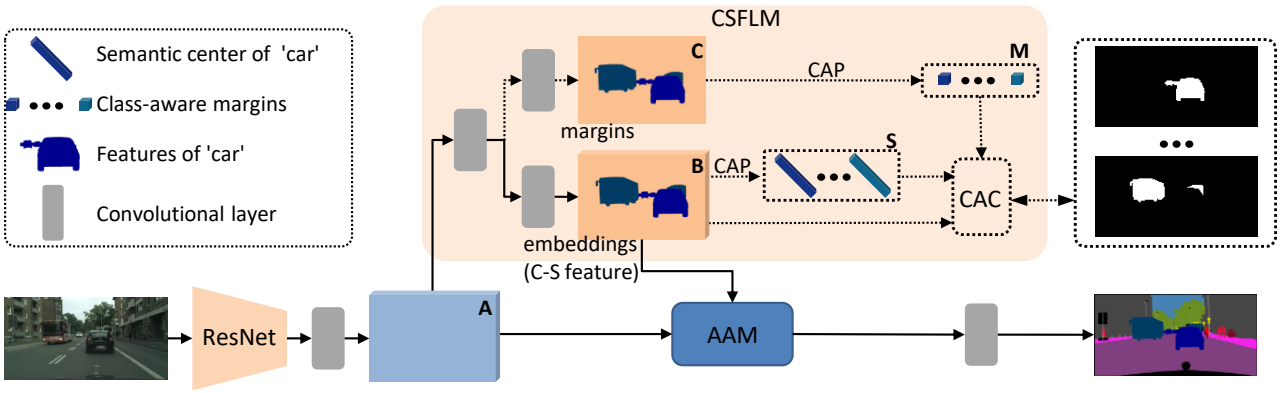

Figure 1: An overview of the proposed Consistent-Separable Feature Representation Network. The dotted line denotes that it is only used during training phase. CAP denotes Class-wise Average pooling. CSFLM denotes Consistent-Separable Feature Learning Module. AAM denotes Adaptive feature Aggregation Module. CAC denotes Class-Aware Consistency loss. Embeddings (C-S feature) denotes that we name it feature embeddings in training phase and C-S feature in inference phase. A, B and C denote features. Note that we only show the two categories of 'car' and 'bus' for brevity in CSFLM.

C-S features into the original features from the backbone to obtain final features for semantic prediction. Moreover, to obtain the better final features, we design an Adaptive feature Aggregation Module (AAM) to fuse the C-S features and original features. We aggregate the features in spatial dimension for both the original features and C-S features, and then we concatenate them in channel dimension to obtain the final feature to predict the label of each pixel. Due to CAC loss force the features among the same class keeping consistency and pull the features of different classes staying apart, our method could build the consistent-separable feature representation and thus can obtain more discriminative feature representation for segmentation.

To verify the effectiveness of the proposed method, we plug our module into recent state-of-the-arts segmentation methods (e.g., Non-local, PSP, Deeplabv3+.), and the results indicate that our method achieves impressive improvements compared with these strong baselines. We further carry out extensive experiments on four competitive datasets, including Cityscapes dataset (Cordts et al. 2016), ADE20K dataset (Zhou et al. 2017), COCO Stuff dataset (Caesar, Uijlings, and Ferrari 2018), and PASCAL Context dataset (Mottaghi et al. 2014) to evaluate the algorithm, and it achieves state-of-the-art performance on above four datasets.

Our main contributions can be summarized as follows:

- We propose a Consistent-Separable Feature Representation Network (CSFRN) to build the discriminative features, which are consistent among the same class and separable between the different classes.
- A Consistent-Separable Feature Learning Module (CSFLM) is designed to obtain Consistent-Separable (C-S) feature with the supervision of the proposed class-aware consistency loss. Moreover, we design an Adaptive feature Aggregation Module (AAM) for fusing the C-S features and the original features, further improving the performance.
- CSFLM provides a plug-and-play module which could be easily employed into existing segmentation networks with negligible cost and achieve significant improvement compared to these networks. Extensive experiments demonstrate that our method achieves state-of-the-art performance on four competitive benchmarks.

Related Work

Semantic Segmentation Fully Convolutional Networks (FCN) (Long, Shelhamer, and Darrell 2015) based methods have made significant progress on semantic segmentation. Some works (Lin et al. 2017; Ding et al. 2020) design the encoder-decoder network to recover the detailed spatial information for improving segmentation quality. Recent works show that exploring the context information could obtain better segmentation performance. The methods of constructing the context information could be divided into multi-scale-based methods and attention-based methods. For the multi-scale-based method, ParseNet (Liu, Rabinovich, and Berg 2015) incorporates global information into the features for improving segmentation performance. PSPNet (Zhao et al. 2017) builds context features through aggregating different-region-based features, which are obtained by pyramid pooling modules. DeepLabv3 (Chen et al. 2017) capture context features by atrous spatial pyramid pooling. For the attention-based methods, some works introduce a self-attention mechanism to build context features. DANet (Fu et al. 2019a), CCNet(Huang et al. 2019) exploit relationships among the features to aggregate the features in spatial/channel dimension.

Affinity Modeling Semantic segmentation could not be regarded as a task of independent prediction for each pixel, but the correlation information between different pixels should fully be considered. Several approaches employ structure information for segmentation. (Krähenbühl and Koltun 2011) propose denseCRF to establish pairwise potentials on all pairs of pixels for refining the segmentation results. Liu *et al.* (Liu et al. 2017b) incorporate high-order relations and

label contexts into Markov Random Field for semantic segmentation. Ke *et al.* (Ke et al. 2018) propose adaptive affinity fields to match the semantic relations between neighbor pixels in the label space. Zhao *et al.* (Zhao et al. 2019) propose RMI Loss to model the dependencies among the pixels in label space by using one pixel and its neighbor pixels to represent these pixels.

Different from the above methods, we build a Consistent-Separable Feature Representation Network to obtain the C-S features, which are inter-class compactness and inter-class separability, improving the segmentation performance.

Proposed Method

In this section, we first introduce the overview pipeline of our network. And then, we describe the details of the Consistent-Separable Feature Learning Module and Adaptive feature Aggregation Module. Finally, we describe the training and testing process of the network.

Overview

In this paper, we propose a Consistent-Separable Feature Representation Network to construct the discriminative feature with intra-class compactness and inter-class separability for improving the segmentation performance.

As illustrated in Figure 1, following previous works (Chen et al. 2018), we first utilize the dilated ResNet as the backbone to extract features and the output stride is $1/8$. Then, the features are fed into the Consistent-Separable Feature Learning Module. In this module, at the training phase, we first build the feature embeddings and margins, which could adaptively adjust the degree of intra-class compactness and inter-class separability for different categories. Then, we build the semantic centers for each category and the class-aware margins through average the values of the same category. Last, we exploit the CAC loss to pull the feature embeddings of different classes staying apart and push the feature embeddings of the same class keeping close. With the supervision of CAC loss in CSFLM, the feature embedding will be inter-class compactness and inter-class separability. At the inference phase, we only generate the feature embeddings and we directly regard the feature embeddings as Consistent-Separable (C-S) features. After obtaining the C-S features, the C-S features and original features are passed through the Adaptive feature Aggregation Module to obtain the final features via feature aggregation in spatial and channel dimensions. At last, the final features are utilized to predict the pixel-level labels of the input image.

Consistent-Separable Feature Learning Module

The softmax loss is not good at reducing the feature variation among the same class. Recent works (Yu et al. 2020b; Hu et al. 2020) perform the self-attention mechanism in class-level, implicitly shrinking the intra-class diversity and thus improving the segmentation performance. Different from these methods, we propose a consistent-separable feature learning module to explicitly build the consistent-separable feature through supervision.

As shown in Figure 1 (CSFLM), given an input feature from last stage of the ResNet $\mathbf{A} \in \mathbb{R}^{C\times H\times W}$, where C denotes the channel dimension and $H \times W$ denotes spatial resolution. We first feed $\mathbf{A}$ into a 3×3 convolutional layer to obtain the feature $\mathbf{A}'$. Then, two 1×1 convolutions are employed to generate the feature embeddings $\mathbf{B} \in \mathbb{R}^{C'\times H\times W}$ and margins $\mathbf{C} \in \mathbb{R}^{1\times H\times W}$.

To obtain consistent-separable features, we introduce the class-aware consistency loss to supervise the feature embeddings $\mathbf{B}$.

Class-Aware Consistency Loss For the semantic segmentation task, a ground truth mask is provided for each image. We can know which pixels belong to the same category. And we could build a "semantic center" (a vector with the same dimension as feature embeddings $\mathbf{B}$) for each category. Then, we simultaneously minimize the distances from the feature embedding of each pixel to its corresponding semantic center and maximize the distances from the feature embedding of each pixel to other semantic centers.

Given an input image and the ground truth, we first obtain the feature embeddings $\mathbf{B}$ by the way mentioned above. And we downsample the ground truth into the same size of the feature embeddings $\mathbf{B}$ by nearest neighbor downsampling, yielding a new ground truth. The categories in the smaller ground truth are represented as $\mathrm{K} = \{1, 2, 3, \cdots, k\}$. For a certain category, we could find the pixels, which belong to the category, in the feature embeddings $\mathbf{B}$. Therefore, we could compute the semantic centers $\mathrm{S} = \{s_0, s_1, s_2, \cdots, s_k\}$ for each category:

$$s_k = \frac{1}{N_k}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}_{ij}^{k}(b_{ij}) \tag{1}$$

where s_k denotes the semantic center of category k. H, W denotes the height and width of the feature embeddings $\mathbf{B}$. N_k denotes the total number of pixels which belong to category k. $\mathbb{1}_{ij}^{k}$ is an indicator function, being 1 if the ground truth label in position (i, j) is k and 0 otherwise. b_{ij} denotes the feature vector of position (i, j) in feature embeddings $\mathbf{B}$.

We construct a Gaussian function to compute the distance between each feature vector in feature embeddings $\mathbf{B}$ and its corresponding semantic center. This function map the distance between the the feature vector b_{ij} and a semantic center s_k into a value ranged in $[0, 1)$. Moreover, we introduce class-aware margins $\mathrm{M} = \{m_0, m_1\, m_2, \cdots, m_k\}$ to adaptively adjust the degree of intra-class compactness and inter-class separability.

$$\phi(b_{ij}, s_k) = 1 - \exp\left(-\frac{\|b_{ij} - s_k\|^2}{2m_k^2}\right) \tag{2}$$

where $\phi(b_{ij}, s_k)$ measures the distance between the feature vector b_{ij} and a semantic center s_k. m_k denotes the margin for the category k. In practice, instead of using the standard $\frac{1}{2m_k^2}$, we make use of $\exp(m_k * \delta)$ with a fix scalar δ to adjust the degree. δ is set to 10 by default. The computation of m_k can be formulated as follows:

$$m_k = \frac{1}{N_k} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}_{ij}^{k}(c_{ij}) \tag{3}$$

where c_{ij} denotes the element of position (i, j) in margins $\mathbf{C}$. m_k could be regarded as the specific margin for category k.

To achieve intra-class consistency and inter-class separability, if the semantic center of feature vector b_{ij} is s_k, $\phi(b_{ij}, s_k)$ should be 0, otherwise 1. Therefore, the Class-Aware Consistency loss function can be defined as follows:

$$L_{cac} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N} \sum_{i=1}^{H} \sum_{j=1}^{W} \{\mathbb{1}_{ij}^{k}(\mathcal{L}(d, 0)) + \mathbb{1}_{ij}^{nok}(\mathcal{L}(d, 1))\} \tag{4}$$

where $d = \phi(b_{ij}, s_k)$, $\mathcal{L}(\cdot, \cdot)$ denotes a binary classification loss function. In the class-aware consistency loss, the first term $\mathbb{1}_{ij}^{k}(\mathcal{L}(d, 0))$ means to force the feature vector in the feature embeddings $\mathbf{B}$ close to its corresponding semantic center. The second term $\mathbb{1}_{ij}^{nok}(\mathcal{L}(d, 1))$ force the feature vector to be farther away from other semantic centers. Considering that the number of pixels belonging to a certain class in an image is much less than the number of pixels not belonging to this class, there will be a problem of class imbalance. In practice, we opt for using Lovasz-hinge loss (Yu and Blaschko 2015; Neven et al. 2019) as the binary classification function. Lovasz-hinge loss measures the overlap between predictions and ground truths and is insensitive to the number of foreground/background pixels, thus alleviating the class-imbalance problem.

By minimizing the loss function, we could obtain consistent-separable features with intra-class compactness and inter-class separability, improving the discriminative ability of the feature representation and thus achieving better segmentation performance.

Smooth Loss As mentioned above, we calculate a margin for each category via Eq. (3) to adaptively adjust the degree of intra-class compactness and inter-class separability for different categories. The margins predicted by the pixels belonging to the same category should be as consistent as possible. Therefore we add a smooth loss as follows:

$$L_{smooth} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}_{ij}^{k}(\|c_{ij} - m_k\|^2) \tag{5}$$

Adaptive Feature Aggregation Module

To further improve the ability of feature representation, we design an adaptive feature aggregation module to fuse the C-S features and original features from the backbone. Deformable convolution (Dai et al. 2017) can adaptively aggregate features in spatial dimension through learnable offsets. Since the feature of each pixel of the C-S features is close to its corresponding semantic center, it contains accurate category information. Thus, the C-S features could guide the generation of the offsets for deformable convolution.

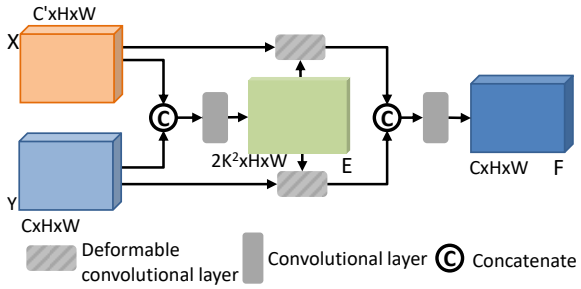

Figure 2: Details of Adaptive feature Aggregation Module.

As illustrated in Figure 2, given a C-S feature $\mathbf{X}$ obtained by CSFLM, and an original feature $\mathbf{Y}$ obtained from the backbone, we first concatenate them in channel dimension. After that, the concatenated features are passed through a 3×3 convolutional layer to predict the offsets $\mathbf{E} \in \mathbb{R}^{2K^2 \times H \times W}$ for deformable convolution operation. K is the kernel size of the deformable convolutional layer. After obtaining the offsets, we aggregate the feature in spatial dimension for both C-S features $\mathbf{X}$ and original features $\mathbf{Y}$ through a 3×3 deformable convolutional layer.

$$\mathbf{X}' = W_a(\mathbf{X}, \mathbf{E}), \mathbf{Y}' = W_b(\mathbf{Y}, \mathbf{E}) \tag{6}$$

where the $W_a(\cdot, \cdot)$ and $W_b(\cdot, \cdot)$ are the deformable convolutional layer. Then, we concatenate the spatial aggregated features $\mathbf{X}'$ and $\mathbf{Y}'$ in channel dimension and employ a 1×1 convolutional layer to generate the final feature $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$. And the final feature $\mathbf{F}$ is employed to predict the class of each pixel of the input image.

Training & Inference

Training We jointly learn the segmentation prediction and consistent-separable features together with the fusion module in an end-to-end fashion. Here, the segmentation prediction is supervised by standard cross-entropy loss, noted as L_{seg}. Therefore, the total loss L is computed as:

$$L = \lambda_1 L_{seg} + \lambda_2 L_{cac} + \lambda_3 L_{smooth} \tag{7}$$

where λ_1, λ_2 and λ_3 are hyper-parameters that control the weighting among the three losses. In practice, we set $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 0.2$.

Inference During inference, we only generate the feature embeddings and directly regard the feature embeddings as consistent-separable features in Consistent-Separable Feature Learning Module. After obtaining the C-S features, we utilize the adaptive feature aggregation module to fuse the C-S features and original features to generate final features for predicting the label of each pixel.

Experiments

To evaluate our proposed method, we carry out extensive experiments on the Cityscapes dataset, ADE20K dataset, COCO Stuff dataset and PASCAL Context dataset. For these datasets, we use the mean Intersection over Union(mIoU) as an evaluation criterion. Experimental results demonstrate

Methods	mIoU	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
PSP	78.65	98.0	84.5	92.8	59.2	60.7	64.9	71.4	79.5	92.6	66.6	94.9	82.7	64.2	95.3	73.6	87.0	80.7	68.2	77.8
+CSFLM	**80.03**	98.1	84.9	93.1	63.1	62.2	65.3	73.2	81.3	92.7	66.8	95.0	83.5	66.2	95.4	77.1	89.3	84.7	69.6	78.9
Nonlocal	77.64	98.0	84.3	92.7	57.0	61.3	64.8	71.3	80.1	92.6	65.0	94.8	82.0	60.8	95.1	74.7	85.9	75.7	61.3	77.8
+CSFLM	**78.91**	98.1	85.0	93.0	60.7	62.0	65.5	72.8	81.2	92.6	66.0	94.9	82.8	62.4	95.4	79.6	88.3	77.1	63.7	78.6
Deeplabv3+	78.5	98.1	85.0	93.1	56.4	62.1	67.1	72.1	80.7	92.7	65.6	95.0	83.3	65.0	95.3	73.9	86.2	73.9	67.4	78.2
+CSFLM	**79.87**	98.2	86.0	93.2	58.6	62.8	68.0	73.5	81.8	92.8	67.3	95.1	83.8	66.0	95.5	80.6	88.1	77.5	69.6	79.3

Table 1: Category-wise results on Cityscapes validation set. Our method improves all strong baselines in Mean IoU, and improve baselines by a significant margin in some categories, such as "truck", "train", "bus", etc. The score of each category retain one decimal place for brevity.

that our method achieves state-of-the-art performance on the above four benchmarks. In the following section, we first conduct ablation studies to verify the effectiveness of our method. And then, we perform the visualization analysis. Finally, we compare our method with state-of-the-arts.

Datasets and Implementation Details

Cityscapes It contains 5000 high quality pixel-level annotated images. And these annotated images are divided into 2975, 500 and 1525 images for training, validation, and testing. The dataset contains 19 classes and each image is of 1024×2048 resolution.

ADE20K There are 20K images in the training set, 2K images in the validation set, and 3K images in the testing set. Images in this dataset are labeled as 150 classes, including 35 stuff concepts and 115 discrete objects.

COCO Stuff The COCO Stuff dataset contains 10,000 images, including 9,000 images for training and 1,000 images for testing. Following (Li et al. 2019), we report our results on 171 categories.

PASCAL Context There are 4998 images for training and 5105 images for testing. Following (Yuan, Chen, and Wang 2019; Yu et al. 2020b), we evaluate our method on the most frequent 59 classes.

Implementation Details The PyTorch framework is employed to implement our network. We use the dilated ResNet as our backbone in the following experiments, and all backbones are pretrained on the ImageNet dataset (Deng et al. 2009). During training phase, the initial learning rate is set to 0.01 for the Cityscapes dataset with the momentum of 0.9, the weight decay of 0.0001 and the batch size of 8. Following (Zhang et al. 2018), the 'poly' learning policy is used to decay the initial learning rate by multiplying $(1-\frac{iter}{total\_iter})^{0.9}$ after each iteration. For data augmentation, random horizontal filp, random cropping (cropsize 769) and random resizing with scales range [0.75, 2.0] are employed in the ablation study. Besides, we train the model with Synchronized Batch Normalization for 40k iterations for ablation study, and 90k iterations for submission to the server. During the testing phase, following (Zhang et al. 2018), the sliding-window method is used for evaluation.

Ablation Study for CSFLM

In order to analyze and verify the effectiveness of CSFLM, for the experiments in this subsection, we use ResNet50 (He et al. 2016) as the backbone, and we directly concatenate the C-S features obtained from CSFLM into the original features without adaptive feature aggregation module.

Improvements upon Strong Baselines To verify the effectiveness and generality of the CSFLM, we apply it to various state-of-the-art networks, including PSPNet (Zhao et al. 2017), Non-Local (Chen et al. 2017) and Deeplabv3+ (Chen et al. 2017). We plug CSFLM after PPM, Non-local block and ASPP in PSP, Non-local and Deeplabv3+ respectively. The experimental results are shown in Table 1. We report the category-wise performance for each method. It can be seen that our method achieves continuous improvements on various state-of-the-art segmentation models.

Complexity Analysis We further analyze the efficiency of our method. We measure the parameters and GFLOPs to investigate the complexity of the network. All experiments adopt the ResNet-50 as the backbone. As illustrated in Table 2, our method improves the baselines by 1.27% - 3.09% in MeanIoU, and only increase by 4.5% - 7.3% and about 2.6M in terms of GFLOPs and parameters. Thus, our method is light-weighted and generality for segmentation networks.

The Influence of Channel Dimension In CSFLM, we introduce the feature embeddings to obtain the C-S features through optimizing the class-aware consistency loss. Here, we conduct more experiments to explore the importance of the feature embeddings dimensions. As shown in Table 3, we model the feature embeddings with channels of 8, 16, 32, and 64 in CSFLM. It can be seen that the CSFLM is not sensitive to the dimension of the feature embeddings. Furthermore, we also report the GFOLPs to investigate the computation complexity with different dimensions.

Effectiveness of the CAC loss For CAC loss, we simultaneously minimize the distances from the feature embedding of each pixel to its corresponding class semantic center and maximize the distances from the feature embedding of each pixel to other class semantic centers. To validate the effectiveness of this supervision manner, we use Lovasz-Softmax (Yu and Blaschko 2015) to directly supervise the

Method	mIoU%	GFLOPs	Parameters(M)
FCN	71.32	149.40	37.70
+CSFLM	74.41(↑3.09)	160.28(↑7.3%)	40.36(↑2.66)
PSP	78.65	178.48	48.98
+CSFLM	80.03(↑1.38)	189.36(↑6.1%)	51.63(↑2.65)
Nonlocal	77.64	239.88	50.02
+CSFLM	78.91(↑1.27)	250.75(↑4.5%)	52.68(↑2.66)
Deeplabv3+	78.49	176.36	43.59
+CSFLM	79.87(↑1.38)	187.24(↑6.2%)	46.25(↑2.66)

Table 2: Complexity comparison. When computing GFLOPs, the resolution of the input image is 512×512.

Method	Channel	GFLOPs	mIoU%
Deeplabv3+	-	176.36	78.49
+CSFLM	8	187.14	79.57
+CSFLM	16	187.17	79.78
+CSFLM	32	187.24	**79.87**
+CSFLM	64	187.37	79.79

Table 3: Ablation study on channel dimensions in CSFLM.

Method	Supervision Manner	Loss	mIoU%
Deeplabv3+	-	-	78.49
+Aux Loss	Mask	L-Softmax	78.11
+CSFLM	Distance	CAC(BCE)	78.70
+CSFLM	Distance	CAC(LS)	**79.87**

Table 4: Ablation study on supervision on CSFLM.

Method	Backbone	mIoU%
Deeplabv3+	ResNet50	78.49
+CSFLM w/o margins	ResNet50	79.33
+CSFLM w/ margins	ResNet50	**79.87**

Table 5: Ablation study on class-aware margins in CSFLM.

category mask (Like adding an auxiliary supervisor) instead of using CAC loss to supervise distance. As shown in Table 4, we use deeplabv3+ with resnet50 as our baseline. CAC (BCE) denotes that we use binary cross-entropy loss as the binary classification loss in Eq. (4). CAC (LS) denotes that we use Lovasz-hinge loss in Eq. (4) to alleviate the problem of foreground/background imbalance. Our method could achieve a better result. It indicates that our supervision manner is effective for segmentation performance.

Effectiveness of the Class-aware Margins As shown in Table 5, we verify the effectiveness of the class-aware margins in CSFLM. It can be seen that building the class-aware margins could achieve a better result.

Ablation Study for AAM

Effectiveness of AAM To verify the effectiveness of AAM, we apply the AAM on the network after CSFLM. As illustrated in Table 6, We use the deeplabv3+ as the baseline, which achieves 78.49% in mean IoU. After applying the CSFLM, we achieve 79.87% in Mean IoU. Further, we add the AAM after the CSFLM, our method achieves 80.18% and improves by a large margin over the baseline.

Method	Backbone	mIoU%
Deeplabv3+	ResNet50	78.49
+CSFLM	ResNet50	79.87
+CSFLM&AAM	ResNet50	**80.18**
+CSFLM&AAM	ResNet101	81.21
+CSFLM&AAM&MG	ResNet101	81.46
+CSFLM&AAM&MG&OHEM	ResNet101	81.80
+CSFLM&AAM&MG&OHEM&MS	ResNet101	**82.91**

Table 6: Ablation study on roles of each components.

Improvements Strategies As illustrated in Table 6, our method could achieve 81.21% in MeanIoU with resnet101. Moreover, following (Fu et al. 2019a; Chen et al. 2018), we adopt some strategies for improving the performance. (1) Multi-grid(MG) (Chen et al. 2017): The multi-dilations with different sizes $\{4,8,16\}$ are employed in last ResNet block. With MG, our method obtain 81.46% in Mean IoU. (2) OHEM, we use OHEM (Yuan and Wang 2018) in L_{seg}, which improves the performance to 81.80%. (3) Multi-scale(MS): We adopt 6 image scales $\{0.25, 0.5, 1, 1.25, 1.5, 1.75\}$. Our method achieves 82.91% in mIoU on the Cityscapes val set with multi-scale inference.

Visual Analysis

Visualization on C-S features To verify whether the learned C-S features are intra-class compactness and inter-class separability, we visualize the C-S features. As shown in Figure 3, we randomly select three query points in input images (a), which are marked by red, blue and green circles. And, we calculate the distance between the feature of the query point and all other points though Eq. (2). Figure 3 (b) is the distance map with red circle as the query point. The category of this point is "road" in the first row of Figure 3 . It can be seen that features of points belonging to the "road" category are relatively consistent and relatively far away from the features of other categories. Figure 3 (c) and (d) use green and blue circle points as query points, respectively. From Figure 3 (c) and (d), it also can be seen that features of the same category are relatively consistent, while features of different categories are far apart.

Comparison of Visualization Results As illustrated in Figure 4, we visualize the prediction results of our method over strong baselines. And we use the red box to highlight the region of improvement through our method. It can be seen that some misclassified categories are now correctly classified, such as "wall", "truck", etc.

Comparison with State-of-the-Arts

To prove the generality of our proposed method, we perform more experiments on four datasets. Our model is based on DeepLabv3+ and adopt ResNet101 as the backbone.

Results on Cityscapes Dataset We compare our method with state-of-the-arts on the Cityscapes testing set. For fair comparision, we train our method with ResNet101 on fine annotated data. Experimental results are shown in Table 7.

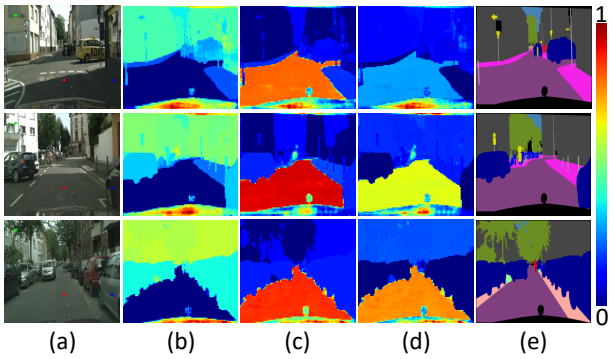

Figure 3: Distance map visualization. (a) input image, and we mark three points with red, blue and green circles. (b) distance map with red circle as the query point. (c) distance map with green circle as the query point. (d) distance map with blue circle as the query point. (e) ground truth.

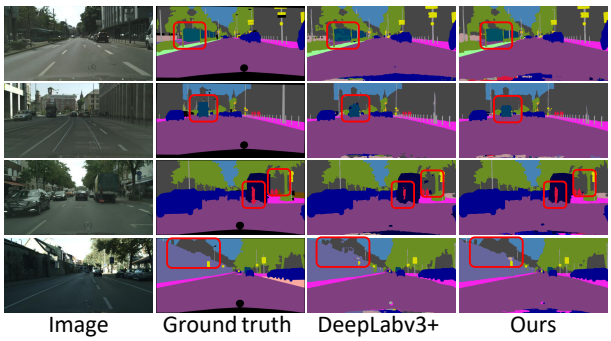

Figure 4: Visualization results compared to Deeplabv3+.

Method	Backbone	mIoU%
DFN (Yu et al. 2018)	ResNet101	79.3
DANet (Fu et al. 2019a)	ResNet101	81.5
ANNet (Zhu et al. 2019)	ResNet101	81.3
BFP (Ding et al. 2019a)	ResNet101	81.4
ACF (Zhang et al. 2019a)	ResNet101	81.9
CPNet (Yu et al. 2020b)	ResNet101	81.3
SPNet (Hou et al. 2020)	ResNet101	82.0
Ours	ResNet101	**82.6**

Table 7: Segmentation results on Cityscapes testing set.

Our method achieves 82.6% mIoU, which outperforms recent existing models. Compared with recent SPNet (Hou et al. 2020), we achieve better performance.

Results on ADE20K Dataset We further carry out experiments on the ADE20K dataset to verify the generality of our method. Following (Zhang et al. 2018; Fu et al. 2019a), we adopt data augmentation with random scaling in training phase and multi-scale in inference phase. Results are shown in Table 8. Our CSFRN achieves state-of-the-art performance 46.65% in MeanIoU on the validation set.

Results on COCO Stuff Dataset We also conduct the experiments on the COCO Stuff dataset to further evaluate the performance of our method. We adopt the same training and testing strategy as the ADE20K dataset. The comparison is shown in Table 9. Our method obtains 41.3% in Mean IoU, which outperforms previous methods by a large margin.

Method	Backbone	mIoU%
EncNet (Zhang et al. 2018)	ResNet101	44.65
APCNet (He et al. 2019)	ResNet101	45.38
CFNet (Zhang et al. 2019b)	ResNet101	44.89
ANNet (Zhu et al. 2019)	ResNet101	45.24
SPNet (Hou et al. 2020)	ResNet101	45.60
RGNet (Yu et al. 2020a)	ResNet101	45.80
DRANet (Fu et al. 2020)	ResNet101	46.18
Ours	ResNet101	**46.65**

Table 8: Segmentation results on ADE20K validation set.

Method	Backbone	mIoU%
RefineNet (Lin et al. 2017)	ResNet101	33.6
DANet (Fu et al. 2019a)	ResNet101	39.7
SVCNet (Ding et al. 2019b)	ResNet101	39.6
EMANet (Li et al. 2019)	ResNet101	39.9
ACNet (Fu et al. 2019b)	ResNet101	40.1
SpyGR (Li et al. 2020)	ResNet101	39.9
Ours	ResNet101	**41.3**

Table 9: Segmentation results on COCO Stuff testing set.

Method	Backbone	mIoU%
PSPNet (Zhao et al. 2017)	ResNet101	47.8
BFP (Ding et al. 2019a)	ResNet101	53.6
DANet (Fu et al. 2019a)	ResNet101	52.6
HRNet (Sun et al. 2019)	HRNetV2-W48	54.0
SpyGR (Li et al. 2020)	ResNet101	52.8
CPNet (Yu et al. 2020b)	ResNet101	53.9
SPNet (Hou et al. 2020)	ResNet101	54.5
Ours	ResNet101	**55.9**

Table 10: Segmentation results on Pascal Context testing set.

Results on PASCAL Context Dataset We further compare our method with existing methods on the PASCAL Context dataset. Following (Fu et al. 2019a; Yu et al. 2020b), we adopt data augmentation with random scaling and multi-testing in the training and testing phase. Quantitative results are shown in Table 10. Our CSFRN achieves 55.9% in Mean IoU, which outperforms previous state-of-the-art methods.

Conclusion

In this paper, we have presented a Consistent-Separable Feature Representation Network (CSFRN) to obtain the discriminative features for per-pixel semantic prediction. Specifically, we designed a Consistent-Separable Feature Learning Module to obtain the Consistent-Separable (C-S) features via the proposed Class-Aware Consistency Loss. Moreover, we developed an Adaptive feature Aggregation Module to fuse the original features and C-S features for better performance. Extensive experimental results on Cityscapes, ADE20K, COCO Stuff and PASCAL Context have demonstrated that our method achieves state-of-the-art performance on these challenging datasets.

Acknowledgements

This work was supported by National Natural Science Foundation of China (61922086, 61872366) and Beijing Natural Science Foundation (4192059, JQ20022).

References

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1209–1218.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* .

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 764–773.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ding, H.; Jiang, X.; Liu, A. Q.; Thalmann, N. M.; and Wang, G. 2019a. Boundary-aware feature propagation for scene segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 6819–6829.

Ding, H.; Jiang, X.; Shuai, B.; Liu, A. Q.; and Wang, G. 2019b. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8885–8894.

Ding, H.; Jiang, X.; Shuai, B.; Liu, A. Q.; and Wang, G. 2020. Semantic segmentation with context encoding and multi-path decoding. *IEEE Transactions on Image Processing* 29: 3520–3533.

Fu, J.; Liu, J.; Jiang, J.; Li, Y.; Bao, Y.; and Lu, H. 2020. Scene Segmentation With Dual Relation-Aware Attention Network. *IEEE Transactions on Neural Networks and Learning Systems* .

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019a. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3146–3154.

Fu, J.; Liu, J.; Wang, Y.; Li, Y.; Bao, Y.; Tang, J.; and Lu, H. 2019b. Adaptive context network for scene parsing. In *Proceedings of the IEEE international conference on computer vision*, 6748–6757.

He, J.; Deng, Z.; Zhou, L.; Wang, Y.; and Qiao, Y. 2019. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7519–7528.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, Q.; Zhang, L.; Cheng, M.-M.; and Feng, J. 2020. Strip Pooling: Rethinking Spatial Pooling for Scene Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4003–4012.

Hu, H.; Ji, D.; Gan, W.; Bai, S.; Wu, W.; and Yan, J. 2020. Class-wise Dynamic Graph Convolution for Semantic Segmentation. *arXiv preprint arXiv:2007.09690* .

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 603–612.

Ke, T.-W.; Hwang, J.-J.; Liu, Z.; and Yu, S. X. 2018. Adaptive affinity fields for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 587–602.

Krähenbühl, P.; and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, 109–117.

Li, X.; Yang, Y.; Zhao, Q.; Shen, T.; Lin, Z.; and Liu, H. 2020. Spatial Pyramid Based Graph Reasoning for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8950–8959.

Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; and Liu, H. 2019. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 9167–9176.

Lin, G.; Milan, A.; Shen, C.; and Reid, I. 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1925–1934.

Liu, W.; Rabinovich, A.; and Berg, A. C. 2015. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579* .

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017a. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Liu, Z.; Li, X.; Luo, P.; Loy, C. C.; and Tang, X. 2017b. Deep learning markov random field for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence* 40(8): 1814–1828.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 891–898.

Neven, D.; Brabandere, B. D.; Proesmans, M.; and Gool, L. V. 2019. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8837–8845.

Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; and Wang, J. 2019. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514* .

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018a. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25(7): 926–930.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Yu, C.; Liu, Y.; Gao, C.; Shen, C.; and Sang, N. 2020a. Representative Graph Neural Network. In *European Conference on Computer Vision*, 379–396.

Yu, C.; Wang, J.; Gao, C.; Yu, G.; Shen, C.; and Sang, N. 2020b. Context Prior for Scene Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12416–12425.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1857–1866.

Yu, J.; and Blaschko, M. 2015. Learning submodular losses with the Lovász hinge. In *International Conference on Machine Learning*, 1623–1631.

Yuan, Y.; Chen, X.; and Wang, J. 2019. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065* .

Yuan, Y.; and Wang, J. 2018. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916* .

Zhang, F.; Chen, Y.; Li, Z.; Hong, Z.; Liu, J.; Ma, F.; Han, J.; and Ding, E. 2019a. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 6798–6807.

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7151–7160.

Zhang, H.; Zhang, H.; Wang, C.; and Xie, J. 2019b. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 548–557.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, S.; Wang, Y.; Yang, Z.; and Cai, D. 2019. Region mutual information loss for semantic segmentation. In *Advances in Neural Information Processing Systems*, 11117–11127.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; and Bai, X. 2019. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 593–602.