Object-Contextual Representations for Semantic Segmentation

Yuhui Yuan[1,2,3], Xilin Chen[2,3], Jingdong Wang[1]
[1]Microsoft Research Asia
[2]Institute of Computing Technology, CAS
[3]University of Chinese Academy of Sciences
{yuhui.yuan, jingdw}@microsoft.com, xlchen@ict.ac.cn

Abstract

In this paper, we address the problem of semantic segmentation and focus on the context aggregation strategy for robust segmentation. Our motivation is that the label of a pixel is the category of the object that the pixel belongs to. We present a simple yet effective approach, object-contextual representations, characterizing a pixel by exploiting the representation of the corresponding object class. First, we construct object regions based on a feature map supervised by the ground-truth segmentation, and then compute the object region representations. Second, we compute the representation similarity between each pixel and each object region, and augment the representation of each pixel with an object contextual representation, which is a weighted aggregation of all the object region representations according to their similarities with the pixel. We empirically demonstrate that the proposed approach achieves competitive performance on six challenging semantic segmentation benchmarks: Cityscapes, ADE20K, LIP, PASCAL VOC 2012, PASCAL-Context and COCO-Stuff. Notably, we achieved the 2nd place on the Cityscapes leaderboard with a single model.

1. Introduction

Semantic segmentation is a problem of assigning a class label to each pixel for an image. It is a fundamental topic in computer vision and is critical for various practical tasks such as autonomous driving and virtual reality. Contextual representations have been shown very important for semantic segmentation in previously extensive works such as DeepLabv3 [4], PSPNet [59] and DANet [13].

Our interest in this paper lies in contextual representation. We start from the fact that the class label assigned to each pixel is the category of the object that the pixel belongs to. We conduct a simple empirical study as follows. We first compute the representation for each object class, by aggregating the representations of all the pixels belonging to the same object class (available from the ground-truth) in an image through average pooling. Then we augment the representation for each pixel with an additional object representation, i.e., the representation of the object that the pixel belongs to. Last, we predict the pixel-wise labels from the augmented representations. The results (our*) are shown in Figure 1. We can see that such an augmentation leads to dramatic performance improvement over the baseline on all the four different benchmarks. e.g., our* improves baseline by 13% on Cityscapes val set.

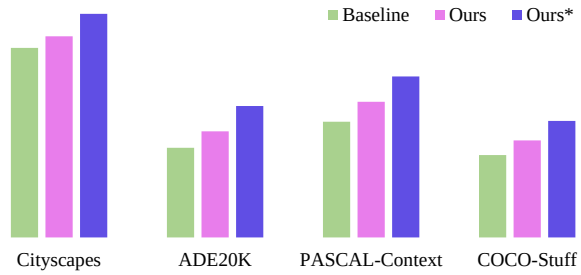


Figure 1: Ours⋆ estimates the object-contextual representations by exploiting the ground-truth segmentation. Ours is the performance of our proposed object-contextual representations. The three methods, ours⋆, ours and the baseline, use the dilated ResNet-101-FCN with output stride 8 as the backbone. We illustrate their results as following: Cityscapes: 75.8%, 79.6%, 88.8%; ADE20K: 39.7%, 44.3%, 54.3%; PASCAL-Context: 45.8%, 53.3%, 63.7%; COCO-Stuff: 32.6%, 38.4%, 46.1%.

Motivated by the above empirical results, we propose a simple yet effective object-contextual representation approach for semantic segmentation. We generate a set of object regions from a feature map, which is supervised by the ground-truth segmentation. Each object region records the probability that every pixel belongs to the corresponding object class, as we do not have the ground-truth segmentation. We estimate the object region representation by aggregating the pixel-wise representations according to the object region of each object class. We augment the representation of each pixel with the object contextual represen-

tation, which is the weighted aggregation of all the object region representations according to their similarities with the corresponding pixel.

We evaluate the proposed approach on six challenging semantic segmentation benchmarks. Our approach outperforms the previous state-of-the-art methods (e.g., PSPNet, DeepLabv3, DANet) by a large margin while being much more efficient. e.g., our approach achieves 83.7% on Cityscapes `test`, 45.66% on ADE20K, 56.65% on LIP, 84.5% on PASCAL VOC 2012, 56.2% on PASCAL-Context and 40.5% on COCO-Stuff.

2. Related Work

Context for segmentation Capturing long-range context information is one of the keys for the current state-of-the-art semantic segmentation approaches [4, 33, 59, 48]. We can partition the previous context-based methods to three kinds following [55]: *nearby spatial context*: ParseNet [33] and PSPNet [59] treated all the pixels over the whole image or a sub-region as the context for the pixels belonging to it. *sampled spatial context*: DeepLabv3 [4] applied multiple atrous convolutions with different atrous rates to capture spatial pyramid context information and regarded these spatially regularly sampled pixels as the context. *attention-based context*: the self-attention [31, 46, 19] mechanism or the non-local scheme [48] proposed to use the weighted combination of all the pixels in the image as the context.

Our work is different from the above works as we define the set of pixels belonging to the same object category as our context (*or* object region) following [55]. We estimate the object region based on a feature map supervised by the ground-truth segmentation, which is much more efficient than OCNet [55].

Graph-based segmentation. The recent works [6, 7, 28, 26, 54] all proposed graph-based approaches to address the semantic segmentation tasks including the double attention [6] , the symbolic graph reasoning layer (SGR) [28] and the graph convolutional unit (GCU) [26] For example, the double-attention [6] first computed a set of global representations and then represented each pixel with the weighted sum of these global representations.

Our approach is related to the above methods but different in the following aspects. Their global representations are different from our object representations as [6, 28, 26] failed associate them with semantic classes and [54] associated them with the classes from a prior knowledge graph and mainly investigates the help of the external human knowledge. Especially, we compare our approach with double-attention by visualizing the predicted attention maps. Besides, our approach proposes a different mechanism to model the relations between the pixels and the object regions while the previous works [28, 26] mainly focused on the relation reasoning between several grouped regions that lack of category-wise semantic meaning. Our approach empirically outperforms all the previous methods by a large margin on various benchmarks.

Region-based segmentation. Before the era of deep learning, region-based approaches [1, 2, 17, 16] have been used to address the scene understanding problem. According to [2], the conventional region-based methods first extracted free-form regions following [45], described them with features and trained a region classifier, and they labeled a pixel with the category of the most confident region that contains it. Besides, the recent work [49] proposed a novel mechanism to mine the object regions and apply classification on the object regions for weakly-supervised semantic segmentation problem. The previous works [2, 49] are related to our work but different in principle, our approach uses the representations of the object regions to augment the representation of pixels while the previous works mainly used the classification scores of the free-form regions to improve the classification of the pixels.

Refinement for segmentation. The previous works have proposed various refinement approaches [14, 24, 12, 44] to improve the semantic segmentation performance. e.g., [24] proposed to use multiple models to learn a refined prediction based on the concatenation of the previously predicted coarse segmentation maps and the original image or a high-level feature map iteratively. Our work proposes a different usage of the coarse predictions, and we use the feature map before channel softmax normalization (coarse segmentation maps) to estimate a set of object regions (and representations) and learn to associate these object region representations with each pixel. Especially, we empirically verify the advantage of our approach in the Appendix.

3. Object-Contextual Representation

Semantic segmentation is a problem of assigning one label l_i to each pixel p_i of an image I, where l_i is one of K different classes. The class label l_i is essentially the label of the object which the pixel lies in. Motivated by this, we present an object-contextual representation approach, characterizing each pixel by exploiting the corresponding object representations.

3.1. Approach

Object region and representation. We partition the image I into K object regions $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_K\}$. Each object region $\mathbf{M}_k$ corresponds to the class k, represented by a 2D map, where each entry indicates if the corresponding pixel belongs to the class k or the probability on the class k.

We compute K object regions via applying the spatial softmax normalization on the feature map $\mathbf{L}$ (with K channels) output from a fully-convolutional network (the intermediate feature map from stage-3 of ResNet in our imple-

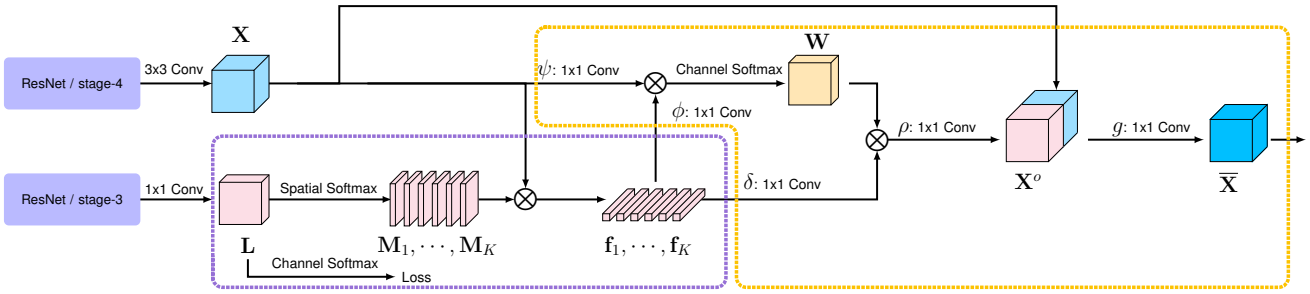


Figure 2: **The pipeline of our approach**. There are two main parts: estimate the object region and representation in the *purple dashed box* and compute the object contextual representation in the *orange dashed box*. We first send $\mathbf{X}$ ($\mathbf{x}_i$ is one element of $\mathbf{X}$) and $\mathbf{L}$ into the *purple dashed box* to compute the object region representations. Then, we send the object region representations and $\mathbf{X}$ into the *orange dashed box* to compute the object contextual representations. Especially, we compute $\mathbf{X}$ and $\mathbf{L}$ based on the output from stage-4 of ResNet and stage-3 of ResNet separately. Refer to the *Architecture* section for more details such as the dimensions of all the representations/ feature maps.

mentation), which are learned under the supervision from the ground-truth segmentation.

We aggregate all the pixel-wise representations weighted by the confidence values from the object regions forming the object region representation,

$$\mathbf{f}_k = \sum_i m_{ki}\mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i$ is the representation of pixel p_i, and m_{ki}, an element of $\mathbf{M}_k$, is the confidence value for pixel p_i.

Object contextual representation. We compute the representation similarity between each pixel and each object region,

$$w_{ik} = \frac{e^{(\phi(\mathbf{f}_k))^\top \psi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{(\phi(\mathbf{f}_j))^\top \psi(\mathbf{x}_i)}}, \tag{2}$$

Here, $\phi(\cdot)$ and $\psi(\cdot)$ are two transformation functions implemented by 1×1 conv $\rightarrow$ BN $\rightarrow$ ReLU. These two functions are inspired by self-attention [46] for a better similarity.

The object contextual representation $\mathbf{x}_i^o$ for pixel p_i is computed as below,

$$\mathbf{x}_i^o = \rho(\sum_{k=1}^{K} w_{ik}\delta(\mathbf{f}_k)), \tag{3}$$

where $\delta(\cdot)$ and $\rho(\cdot)$ are both transformation functions implemented by 1×1 conv $\rightarrow$ BN $\rightarrow$ ReLU, and this follows the bottleneck design in non-local networks [48].

The final representation for pixel p_i is updated as the aggregation of two parts, (1) the original representation $\mathbf{x}_i$, and (2) the object contextual representation $\mathbf{x}_i^o$:

$$\bar{\mathbf{x}}_i = g([\mathbf{x}_i^\top \ (\mathbf{x}_i^o)^\top]^\top). \tag{4}$$

where $g(\cdot)$ is a transform function to fuse the original representation and the object contextual representation, implemented by 1×1 conv $\rightarrow$ BN $\rightarrow$ ReLU. The whole pipeline of our approach is illustrated in Figure 2.

3.2. Architecture

Backbone. We use the ResNet-50/ ResNet-101 pretrained over the ImageNet dataset as the backbone, and make some modifications by following PSPNet [59]: replace the convolutions within the last two blocks by dilated convolutions with dilation rates being 2 and 4, respectively, so that the output stride becomes 8, called dilated ResNet-50/ ResNet-101.

Object region and representation. As illustrated in Figure 2, we estimate the feature maps $\mathbf{L}$ and $\mathbf{X}$ based on the output from the stage-3 of ResNet (of size $H \times W \times 1024$) and the output from the stage-4 of ResNet (of size $H \times W \times 2048$). Then we send both $\mathbf{L}$ and $\mathbf{X}$ to the purple dashed box. The size of $\mathbf{X}$ and $\mathbf{L}$ are $H \times W \times 512$ and $H \times W \times K$ separately. We apply the spatial softmax normalization on $\mathbf{L}$ to estimate the K object regions and each object region $\mathbf{M}_k$ is of size $H \times W$. We also apply channel softmax normalization on $\mathbf{L}$ and supervise it with a pixel-wise loss. Last, we compute the object region representation $\mathbf{f}_k$ by multiplying $\mathbf{X}$ and $\mathbf{M}_k$ according to Equation 1, where each $\mathbf{f}_k$ is a vector of size 512.

Object contextual representation. According to Figure 2, we send $\mathbf{X}$ and the object region representations to the orange dashed box to compute the object contextual representations. First, we apply ψ function on $\mathbf{X}$ and ϕ function on the object region representations separately, and both their channels are decreased from 512 to 256. e.g., $\psi(\mathbf{X})$ is of size $H \times W \times 256$ and $\phi(\mathbf{f}_k)$ is a vector of size 256. Second, we compute the inner product between the reshaped $\psi(\mathbf{X})$ of size $HW \times 256$ and all the object region representations, followed by the channel softmax normalization, output the similarity matrix $\mathbf{W}$ (w_{ik} is one element of $\mathbf{W}$) of size $HW \times K$. Third, we compute the object contextual representation $\mathbf{X}^o$ ($\mathbf{x}_o^i$ is one element of $\mathbf{X}^o$) according to Equation 3, of which the size is $H \times W \times 512$. We use δ function to halve the input channel dimension and ρ

function to double the input channel dimension. Last, we concatenate $\mathbf{X}^o$ and $\mathbf{X}$ and apply g function on it to compute the final representation $\overline{\mathbf{X}}$ ($\bar{\mathbf{x}}_i$ is one element of $\overline{\mathbf{X}}$), of which the size is $H \times W \times 512$.

3.3. Empirical Analysis

We empirically study the results of using the ground-truth object regions and the ground-truth similarities between the pixels and the object regions, called ours*. We evaluate all the related results on the validation set of Cityscapes. We train the ours*, our approach and baseline under the same training and testing settings, such as the dilated ResNet-101 with output stride 8 as the backbone, input crop size 769×769, batch size 8. Especially, we do not apply any tricks such as multi-scale testing, flip testing, online hard sample mining and so on. We further conduct experiment on 3 more benchmarks and summarize the related results in Figure 1.

We illustrate how we apply the ground-truth labels to generate ideal object contextual representation as below: we first generate the K ground-truth object regions by simply setting $m_{ki} = 1$ if the ground-truth label $l_i \equiv k$ and $m_{ki} = 0$ otherwise (m_{ki} is the i_{th} element of $\mathbf{M}_k$). Then we normalize each object region $\mathbf{M}_k$ to ensure the sum of all the confidence values is equal to 1. Second, we set the similarity $w_{ik} = 1$ if the ground-truth label $l_i \equiv k$ and $w_{ik} = 0$ otherwise within Equation 2. According to Figure 1, it can be seen that ours (w/o using ground-truth) benefits from the object contextual representation mechanism and achieves $3.8\%/4.6\%/7.5\%/5.8\%$ absolute improvements over the baseline on Cityscapes/ADE20K/PASCAL-Context/COCO-Stuff respectively.

4. Experiments

4.1. Datasets

Cityscapes. The Cityscapes dataset [8] is tasked for urban scene understanding, which contains 30 classes and only 19 classes of them are used for scene parsing evaluation. The dataset contains $5,000$ high quality pixel-level finely annotated images and $20,000$ coarsely annotated images. The finely annotated $5,000$ images are divided into $2,975/500/1,525$ images for training, validation and testing.

ADE20K. The ADE20K dataset [62] is used in ImageNet scene parsing challenge 2016, which contains 150 classes and diverse scenes with $1,038$ image-level labels. The dataset is divided into 20K/2K/3K images for training, validation and testing.

LIP. The LIP (Look into Person) dataset [15] is employed in the LIP challenge 2016 for single human parsing task, which contains $50,462$ images with 20 classes (19 semantic human part classes and 1 background class). The training, validation, and test sets consist of 30K, 10K, 10K images respectively.

PASCAL VOC. The PASCAL VOC 2012 dataset [11] is one of the most competitive semantic segmentation dataset contains 20 foreground object classes and 1 background class. Most of the images only contain one object in the center area. The original dataset contains $1,464$ train images, $1,449$ val images and $1,456$ test images. [18] has augmented this dataset with extra annotations, resulting in $10,582$ train-aug images.

PASCAL-Context. The PASCAL-Context dataset [36] is a challenging scene parsing dataset contains 59 semantic classes and 1 background class. The training and test set consist of $4,998$ and $5,105$ images respectively.

COCO-Stuff. The COCO-Stuff dataset [36] is a challenging scene parsing dataset contains 59 semantic classes and 1 background class. The training and test set consist of 9K and 1K images respectively.

4.2. Implementation Details

Network. We mainly verify the effectiveness of our approach on ResNet-50 and ResNet-101 We all use the ImageNet-pretrained weights to initialize these models. For both ResNet-50 and ResNet-101, we use the dilated convolution to ensure the output feature map is $8\times$ smaller than the input image if not specified. We have illustrated the details of applying our approach based on ResNet-50 and ResNet-101 in Figure 2. We also extend our OCR on two other architectures including MobileNetV2 and HRNetV2-W48.

Training setting. We illustrate the training settings on all six semantic segmentation datasets as below. *Cityscapes*: we set the initial learning rate as 0.01, weight decay as 0.0005 for, crop size as 769×769 (or 1024×512), batch size as 8 for ResNet series and HRNetV2-W48, batch size as 16 for MobileNetV2. For all the ablation study experiments, we use the 2975 train-fine images as the training set and set the training iterations as 40K/80K/100K for ResNet/HRNetV2-W48/MobileNetV2 respectively if not specified. We only report the performance with MobileNetV2 on Cityscapes following the previous works. *ADE20K*: we set the initial learning rate as 0.02, weight decay as 0.0001, crop size as 520×520, batch size as 16, and 150K training iterations for all the experiments. *LIP*: we set the initial learning rate as 0.007, weight decay as 0.0005, crop size as 473×473, batch size as 32, and 100K training iterations for all the experiments following CE2P [32]. *PASCAL VOC 2012*: we set the initial learning rate as 0.001, weight decay as 0.0001, crop size as 513×513, batch size as 16, and 60K training iterations. *PASCAL-Context*: we set the initial learning rate as 0.001, weight decay as 0.0001, crop size as 520×520, batch size as 16, and 30K training iterations. *COCO-Stuff*: we set the

Table 1: Influence of the object region estimation.

Method	Inserted position	mIoU (%)
Object Region (w/o sup)	stage-3	77.31
Object Region (w/o sup)	stage-4	77.56
Object Region (w/ sup)	stage-3	79.58
Object Region (w/ ground-truth)	stage-3	**83.10**

initial learning rate as 0.001, weight decay as 0.0001, crop size as 520×520, batch size as 16, and 60K training iterations.

Notably, we choose the polynomial learning rate policy with factor $(1 - (\frac{iter}{iter_{max}})^{0.9})$, the weight on the final loss is 1, the weight on the loss used to supervise the object region estimation (*or* auxiliary loss) is 0.4 and the INPLACE-ABN$^{\text{sync}}$ [40] to synchronize the mean and standard-deviation of BN across multiple GPUs. For the data augmentation, we we only apply random flipping horizontally, random scaling in the range of $[0.5, 2]$ and random brightness jittering within the range of $[-10, 10]$. We also choose the same settings for our reproduced PSPNet to ensure the fairness.

4.3. Ablation study

We conduct all of our ablation study experiments with dilated ResNet-101 as our backbone and report all the performance with only single scale testing on the the Cityscapes validation set if not specified.

Object regions. We study the influence of the object region estimation within OCR. First, we remove the supervision on the $\mathbf{L}$ and inject the modified object contextual module (w/o supervision) at different positions including: after stage-3 and after stage-4 of ResNet. We also add an auxiliary loss after stage-3 to ensure the fairness of the experiments. All the other settings are kept the same. We report the related results in Table 1. It can be seen that the direct supervision on $\mathbf{L}$ is crucial for the final performance and the object regions without supervision suffer from poor semantic meaning. Besides, we also investigate the spatial softmax normalization on the $\mathbf{L}$ by removing the spatial softmax before computing the object regions and summarize the results as below: w/o spatial softmax: 78.45% *vs.* w/ spatial softmax: 79.58%. It can be seen that spatial softmax normalization suppresses the background context and highlights the object context, which is the key for better performance. Last, we present the performance with ideal object regions (based on the ground-truth) in Table 1, which is also one of the bottleneck of our approach.

In summary, it can be seen that our approach with coarse object regions already brings significant improvements and we expect higher performance with more accurate object regions. We choose the method Object Region (w/ sup, w/ spatial softmax) as our default setting if not specified.

Representation similarity. We study the influence of the representation similarities between the pixels and the object regions within OCR. We mainly compare our approach with two variant mechanisms. The first mechanism is inspired by the [6], we modify the estimation of similarity w_{ik} in Equation 2 as $w_{ik} = \frac{e^{\theta(\mathbf{x}_i)_k}}{\sum_{j=1}^{K} e^{\theta(\mathbf{x}_i)_j}}$, where $\theta(\mathbf{x}_i)$ is a vector of size K and we implement the function θ with 1×1 conv $\rightarrow$ BN $\rightarrow$ ReLU. The second mechanism directly generates the similarities based on the previously predicted object regions following $w_{ik} = m_{ki}$. We evaluate the above two mechanisms and report their performance as below: 79.01% and 78.02%. It can be seen that our approach (mIoU=79.58%) outperforms both mechanisms. Last, we also report the performance with the ideal similarities (based on the ground truth) as below: 88.01%.

We also illustrate the distribution of the learned similarities in Figure 4. It can be seen that our approach learns reliable similarities for most categories while fails for some object categories such as the traffic-sign. In summary, it can be seen that how to learn accurate similarities is an even more important factor within our approach.

Table 2: Comparison with PSPNet and Self-Attention.

Method	Cityscapes (w/o coarse)	Cityscapes (w/ coarse)	ADE20K	LIP
PSPNet [47]	78.4%	81.2%	43.29%	–
PSPNet (Our impl.)	80.3%	81.6%	44.60%	54.76%
SA (Our impl.)	81.1%	82.0%	44.65%	54.78%
OCR	**81.8%**	**82.4%**	**45.28%**	**55.60%**

Table 3: Comparison with PPM, ASPP and Self-Attention when processing input feature map of size $[1 \times 2048 \times 128 \times 128]$ during inference stage. All of them are evaluated on single P40 GPU with CUDA10.0.

Method	Parameters▲	Memory▲	MAdds ▲	Time▲
PPM [59]	23.1M	792M	379B	99ms
ASPP [4]	15.5M	284M	254B	97ms
Self-Attention	10.5M	2168M	240B	96ms
OCR	10.5M	**202M**	**166B**	**45ms**

Comparison with PSPNet. To extensively compare our OCR with PSPNet, we conduct experiments on three datasets and report the performance on the Cityscapes test set, ADE20K val set and LIP val set in Table 2. We all use the same backbone (ResNet-101) and the same training/testing settings to ensure the fairness. Especially, our reproduced PSPNet outperforms the reported results in [59]. In summary, it can be seen that our OCR outperforms PSPNet on all datatsets.

Comparison with Self-Attention. We compare our OCR with the conventional self-attention approach and report the results in Table 2. It can be seen that our OCR outperforms the conventional self-attention approach on all the evaluated datasets. Notably, we compare their computation complexities to illustrate the advantages of our approach in the following section.

Parameters/Memory/Computation/Time. We compare

Table 4: MobileNetV2 + OCR: Speed (measured by FPS) is tested on P40 GPU with input image of size 1024×512.

Method	FPS	Cityscapes Val. mIoU
MobileNetV2	**31**	69.50%
MobileNetV2 + OCR	28	**74.18%**

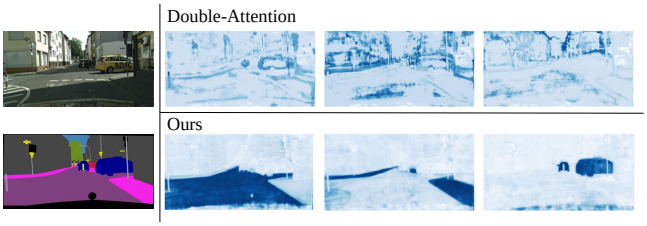
Table 5: Comparison with Double-Attention. K is the number of global representations within Double-Attention. K is exact the number of categories for ours.

Method	K	mIoU (%)
Double-Attention	8	78.52
Double-Attention	16	78.49
Double-Attention	32	78.53
Double-Attention	64	78.65
Double-Attention	128	77.43
OCR	19	**79.58**

the increased parameters, GPU memory, computation cost (measured by the number of Multiply-Adds) and inference time of PPM of PSPNet [59], ASPP of DeepLabv3 [4], Self-Attention and our OCR. We report the results in Table 3 and we evaluate the cost of all the above methods without considering the cost of the backbone (dilated ResNet-101). Especially, we include the cost of the 3×3 convolution that reduce the dimension from 2048 to 512 in our approach to ensure the fairness of the comparison. Considering the parameters of the backbone ResNet-101 is about 42.7M, the increased parameters of both self-attention and our approach are much less compared with the previous PPM [59] and ASPP [4]. Especially, we can see that the self-attention based approaches (e.g., DANet) suffer from much larger GPU memory cost than all the other approaches.

In summary, our OCR is more than **2**$\times$ faster than all the other approaches while requiring less parameters, GPU memory and Multiply-Adds during inference. We argue that our approach is of great practical value for the various tasks that require light-weight solutions. We further apply our OCR based on a light-weight backbone network such as MobileNetV2 and report the performance and the overall speed (include the cost of backbone) in Table 4.

Comparison with Double-Attention. We compare our OCR with the previous Double-Attention [6] that models the relations between the pixel-wise representation and a set of global representations with a different manner. We empirically and visually verify that our OCR is more reliable. We report the related results in Table 5. It can be seen that our approach outperforms the double-attention under various numbers of the global representations. Besides, we also visualize the attention maps (*or* object regions within OCR) of our approach and the ones of Double-Attention in Figure 3 to illustrate why our approach performs better visually. It can be seen that our attention maps (object regions) are of better quality (semantic meaning) than the attention maps learned with Double-Attention.



footnotesize

Figure 3: We randomly choose an image with its label map from Cityscapes val set. The first row illustrates 3 attention maps within Double-Attention and they lack of clear semantic meaning. The second row illustrates 3 attention maps within OCR and they all have specific semantic meaning (e.g., road, sidewalk, car).

4.4. Comparison with State-of-the-art

Table 6: Comparison with baseline on Cityscapes `val`.

Method	Auxiliary Loss	Ms	Flip	mIoU (%)
ResNet-101 FCN	✗	✗	✗	75.31
ResNet-101 FCN	✓	✗	✗	75.80
ResNet-101 FCN + OCR	✓	✗	✗	79.58
ResNet-101 FCN + OCR	✓	✓	✗	80.20
ResNet-101 FCN + OCR	✓	✓	✓	**80.60**

Table 7: Comparison with state-of-the-art on Cityscapes `test`.

Method	Validation	Coarse	Backbone	mIoU (%)
PSPNet [59]	✗	✗	ResNet-101	78.4
PSANet [60]	✗	✗	ResNet-101	78.6
AAF [21]	✗	✗	ResNet-101	79.1
RefineNet [30]	✓	✗	ResNet-101	73.6
SAC [58]	✓	✗	ResNet-101	78.1
DUC-HDC [47]	✓	✗	ResNet-101	77.6
BiSeNet [52]	✓	✗	ResNet-101	78.9
PSANet [60]	✓	✗	ResNet-101	80.1
DFN [53]	✓	✗	ResNet-101	79.3
DSSPN [29]	✓	✗	ResNet-101	77.8
DepthSeg [23]	✓	✗	ResNet-101	78.2
DenseASPP [51]	✓	✗	DenseNet-161	80.6
DANet [13]	✓	✗	ResNet-101	81.5
HRNet [42]	✓	✗	HRNetV2-W48	81.6
PSPNet [59]	✓	✓	ResNet-101	81.2
PSANet [60]	✓	✓	ResNet-101	81.4
DeepLabv3 [4]	✓	✓	ResNet-101	81.3
DeepLabv3+ [5]	✓	✓	Modified Xception	82.1
Mapillary [40]	✓	✓	WideResNet-38	82.0
OCR	✓	✗	ResNet-101	81.8
OCR	✓	✓	ResNet-101	82.4
OCR	✓	✓	HRNetV2-W48	**83.0**

Results on Cityscapes. We first provide a component analysis experiments on the Cityscapes val set and report the results in Table 6. It can be seen that OCR improves 3.78% ↑ over the baseline on Cityscapes val set with single scale testing. Especially, we run all of the experiments for three times and report the mean due to the high variance of the mIoU according to [22].

To compare with the current state-of-the-art methods, We train our model with train-fine and val-fine for 100K iterations. The results are illustrated in Table 7 and it

Table 8: Comparison with the top ranking methods on the Cityscapes leader-board.

Method	class mIoU	class iIoU	category mIoU	class imIoU
Hyundai Mobis AD Lab	**83.8**	**65.0**	**92.4**	82.4
HRNetV2 + OCR (w/ ASP) (Our)	83.7	64.8	**92.4**	**83.5**
iFLYTEK-CV	83.6	64.7	92.1	82.3
VPLR [63]	83.5	64.4	92.2	82.0
HRNetV2 + OCR (Our)	83.3	62.0	92.1	81.7
NV-ADLR	83.2	64.2	92.1	82.2
Tencent AI Lab	82.9	63.9	91.8	80.4
GSCNN [43]	82.8	64.3	92.3	82.7

Table 9: Comparison with state-of-the-art on ADE20K `val`.

Method	Backbone	mIoU (%)
RefineNet [30]	ResNet-101	40.20
RefineNet [30]	ResNet-152	40.70
PSPNet [59]	ResNet-101	43.29
PSPNet [59]	ResNet-152	43.51
SAC [58]	ResNet-101	44.30
PSANet [60]	ResNet-101	43.77
UperNet [50]	ResNet-101	42.66
HRNetV2 [42]	HRNetV2-W48	42.99
DSSPN [29]	ResNet-101	43.68
EncNet [56]	ResNet-101	44.65
SGR [28]	ResNet-101	44.32
GCU [26]	ResNet-101	44.81
CFNet [57]	ResNet-101	44.89
OCR	ResNet-101	45.28
OCR	HRNetV2-W48	**45.66**

can be seen that OCR achieves better performance than all the previous methods based on ResNet-101 (w/o coarse), and we achieve new state-of-the-art performance of 81.8% on the test set and outperforms the DenseASPP based on DenseNet-161 by over 1.0% ↑. We further achieve 82.4% through exploiting the coarsely annotated images. Our approach (w/ coarse) outperforms both DeepLabv3 and DeepLabv3+. Based on a stronger backbone HRNetV2-W48 [42], our OCR achieves higher performance: 83.0%.

In order to compete with the top ranking methods on the Cityscapes leader-board, we further use Mapillary Vistas dataset [37] for training as all the other methods have used this dataset. It can be seen that our approach (HRNetV2 + OCR) achieves very competitive performance w/o using the video information or depth information. We then combine our OCR with ASPP [4] by replacing the global average pooling with our OCR, which (HRNetV2 + OCR (w/ ASP)) achieves 1st on 1 metric and 2nd on 3 of the 4 metrics with only a single model.

Results on ADE20K. We report the comparison with previous state-of-the-art on ADE20K val set in Table 9. It can be seen that our OCR outperforms most of the previous approaches. For example, our OCR (based on ResNet-101) achieves 45.28% while the the conventional self-attention based CFNet [57] achieves 44.89%. We also report the results with HRNetV2-W48: 45.66%.

Results on LIP. We evaluate our OCR on the LIP benchmark and report the related results in Table 10. It can be seen that we achieve the new state-of-the-art performance 55.60% on the validation set of LIP, which outperforms the previous state-of-the-art by 2.5% ↑. We further achieve higher performance 56.65% with stronger backbone HRNetV2-W48. Especially, the human parsing task is different from the previous two scene parsing tasks as it is about labeling each pixel with the part category that it belongs to.

Table 10: Comparison with state-of-the-art on LIP `val`.

Method	Backbone	mIoU (%)
Attention+SSL [15]	ResNet-101	44.73
JPPNet [27]	ResNet-101	51.37
SS-NAN [61]	ResNet-101	47.92
MMAN [35]	ResNet-101	46.81
MuLA [38]	ResNet-101	49.30
CE2P [32]	ResNet-101	53.10
HRNetV2 [42]	HRNetV2-W48	55.90
OCR	ResNet-101	55.60
OCR	HRNetV2-W48	**56.65**

Table 11: Comparison with state-of-the-art on PASCAL VOC 2012 `test`.

Method	Backbone	mIoU (%)
FCN [34]	VGG-16	62.2
DeepLab-CRF [3]	VGG-16	71.6
PSPNet [59]	ResNet-101	82.6
DFN [53]	ResNet-101	82.7
EncNet [56]	ResNet-101	82.9
DANet [13]	ResNet-101	82.6
CFNet [57]	ResNet-101	84.2
OCR	ResNet-101	84.3
OCR	HRNetV2-W48	**84.5**

Table 12: Comparison with state-of-the-art on PASCAL-Context `test`. All the methods are evaluated on 59 classes.

Method	Backbone	mIoU (%)
PSPNet [59]	ResNet-101	47.8
CCL [9]	ResNet-101	51.6
EncNet [56]	ResNet-101	51.7
DANet [13]	ResNet-101	52.6
SVCNet [10]	ResNet-101	53.2
CFNet [57]	ResNet-101	54.0
HRNetV2 [42]	HRNetV2-W48	54.0
OCR	ResNet-101	54.8
OCR	HRNetV2-W48	**56.2**

Results on PASCAL VOC. In Table 11, we illustrate the results on PASCAL VOC 2012 benchmark. We can see that our approach achieves competitive performance on the test set of PASCAL-VOC.

Results on PASCAL-Context. We evaluate our approach on the PASCAL-Context benchmark and report the related results in Table 12. It can be seen that our approach outperforms all the previous state-of-the-art on the test set of PASCAL-Context.

Results on COCO-Stuff. We evaluate our approach on the COCO-Stuff benchmark and report the related results in Table 13. It can be seen that our approach achieves competitive performance on the test set of COCO-Stuff. Especially, we achieve 40.5% based a stronger backbone HRNetV2-48.

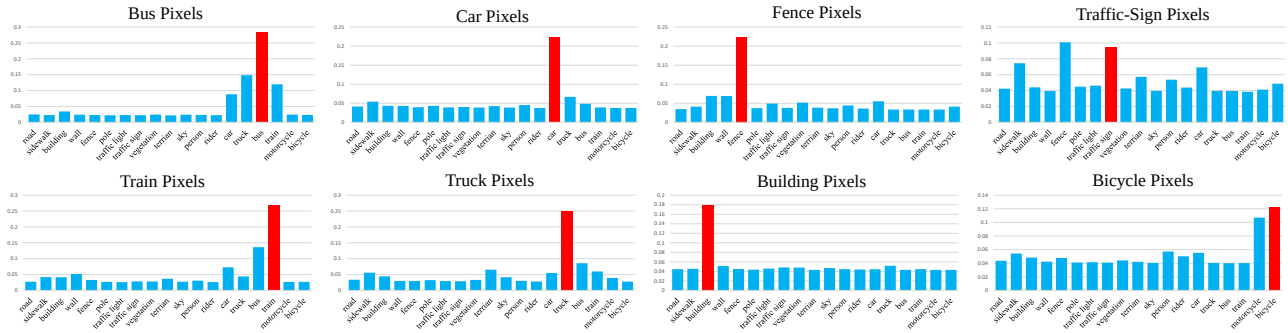


Figure 4: Distribution of the learned similarities between the object regions and the pixels. The x-axis represents 19 object regions and the y-axis represents the accumulated similarities between all the pixels and each object region for the images from Cityscapes val set. It can be seen that the pixels of "Bus", "Car", "Fence", "Train", "Truck" and "Building" distributes most of the similarities on the correct object region, and the distributions of the pixels belonging to "Traffic sign" and "Bicycle" are not as well as the other categories.

Table 13: Comparison with state-of-the-art on COCO-Stuff `test` set.

Method	Backbone	mIoU (%)
FCN [34]	VGG-16	22.7
DAG-RNN [41]	VGG-16	31.2
RefineNet [30]	ResNet-101	33.6
CCL [9]	ResNet-101	35.7
SVCNet [10]	ResNet-101	39.6
DANet [13]	ResNet-101	39.7
OCR	ResNet-101	39.5
OCR	HRNetV2-W48	**40.5**

Visualization. We illustrate the visual improvements of our approach over the baseline on three datasets in this section. We use white dashed boxes to mark the regions that are well-classified by our approach but mis-classified by the baseline. For example, we present the visual results on Cityscapes in Figure 5, the visual results on ADE20K in Figure 6 and the visual results on LIP in Figure 7.

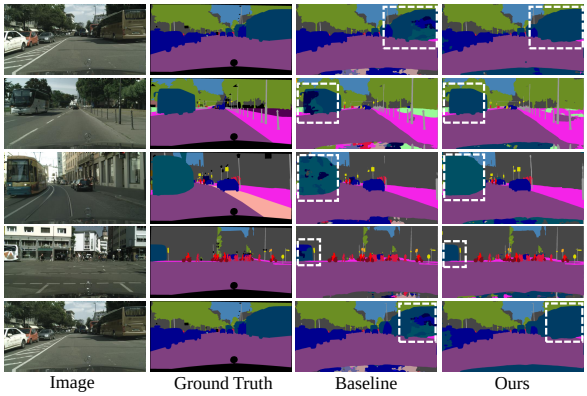


Figure 5: Visual improvements on Cityscapes val set. It can be seen that our approach predicts more accurate segmentation maps for the"Train" and "Bus".

5. Conclusions

In this work, we present the object contextual representations for semantic segmentation through characterizing each pixel with the corresponding object representations. The main reason for the improvement of our approach is that the object contextual representations enable us to address the original pixel-wise labeling problem via a object region labeling mechanism. With the benefit of the object contextual representations, we empirically observe that our approach brings consistent improvements on various datasets.

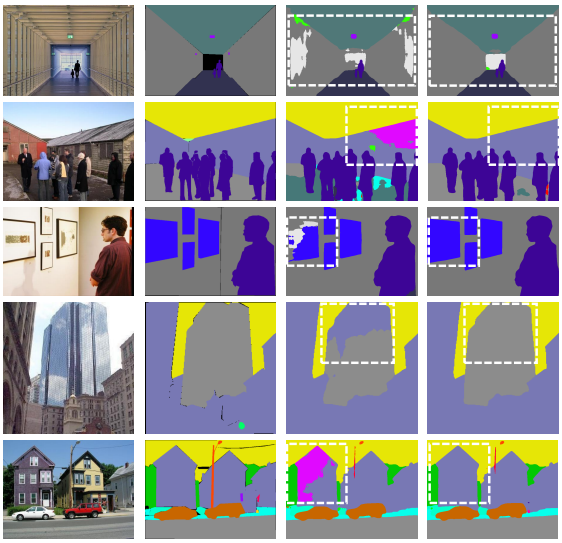


Figure 6: Visual improvements on ADE20K val set. It can be seen that our approach predicts better segmentation maps for the "building" , "wall" and "painting".

References

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. 2012. 2

[2] H. Caesar, J. Uijlings, and V. Ferrari. Region-based semantic segmentation with end-to-end training. In *ECCV*, 2016. 2

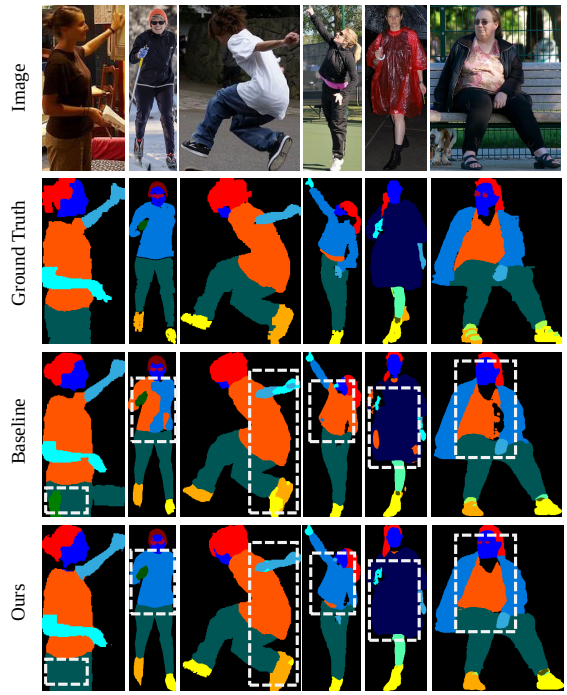


Figure 7: Visual improvements on LIP val set. We arrange each example along columns considering the aspect ratio of human images. It can be seen that our results are close to the ground-truth.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015. 7

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587, 2017. 1, 2, 5, 6, 7

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[6] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. A$\hat{2}$-nets: Double attention networks. In *NIPS*, 2018. 2, 5, 6

[7] Y. Chen, M. Rohrbach, Z. Yan, S. Yan, J. Feng, and Y. Kalantidis. Graph-based global reasoning networks. *arXiv:1811.12814*, 2018. 2

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 4

[9] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, 2018. 7, 8

[10] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, 2019. 7, 8

[11] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 4

[12] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele. Learning to refine human pose estimation. In *CVPRW*, 2018. 2, 11

[13] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *arXiv:1809.02983*, 2018. 1, 6, 7, 8

[14] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *CVPR*, 2017. 2, 11

[15] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017. 4, 7

[16] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 2

[17] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 2

[18] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 4

[19] L. Huang, Y. Yuan, J. Guo, C. Zhang, X. Chen, and J. Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv:1907.12273*, 2019. 2

[20] Y.-H. Huang, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool. Error correction for dense semantic image labeling. In *CVPRW*, 2018. 11

[21] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, 2018. 6

[22] A. Kirillov, R. B. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. CVPR, 2019. 6

[23] S. Kong and C. C. Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *CVPR*, 2018. 6

[24] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *CVPR*, 2016. 2, 11

[25] X. Li, Z. Liu, P. Luo, C. Change Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, 2017. 11

[26] Y. Li and A. Gupta. Beyond grids: Learning graph representations for visual recognition. In *NIPS*, 2018. 2, 7

[27] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *PAMI*, 2018. 7

[28] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing. Symbolic graph reasoning meets convolutions. In *NIPS*, 2018. 2, 7

[29] X. Liang, H. Zhou, and E. Xing. Dynamic-structured semantic propagation network. In *CVPR*, 2018. 6, 7

[30] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 6, 7, 8

[31] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. 2017. 2

[32] T. Liu, T. Ruan, Z. Huang, Y. Wei, S. Wei, Y. Zhao, and T. Huang. Devil in the details: Towards accurate single and multiple human parsing. arXiv:1809.05996, 2018. 4, 7

[33] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. arXiv:1506.04579, 2015. 2

[34] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 7, 8

[35] Y. Luo, Z. Zheng, L. Zheng, G. Tao, Y. Junqing, and Y. Yang. Macro-micro adversarial network for human parsing. In *ECCV*, 2018. 7

[36] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 4

[37] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *CVPR*, 2017. 7

[38] X. Nie, J. Feng, and S. Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, 2018. 7

[39] I. Nigam, C. Huang, and D. Ramanan. Ensemble knowledge transfer for semantic segmentation. In *WACV*, 2018. 11

[40] S. Rota Bul, L. Porzi, and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018. 5, 6

[41] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Scene segmentation with dag-recurrent neural networks. *PAMI*, 2017. 8

[42] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. High-resolution representations for labeling pixels and regions. *arXiv:1904.04514*, 2019. 6, 7

[43] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *ICCV*, 2019. 7

[44] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *PAMI*, 2010. 2, 11

[45] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3

[47] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 6

[48] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 2, 3

[49] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 2

[50] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. 2018. 7

[51] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018. 6

[52] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *ECCV*, 2018. 6

[53] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018. 6, 7

[54] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, and F. Xu. Compact generalized non-local network. In *NIPS*. 2018. 2

[55] Y. Yuhui and W. Jingdong. Ocnet: Object context network for scene parsing. *arXiv:1809.00916*, 2018. 2

[56] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 7

[57] H. Zhang, H. Zhang, C. Wang, and J. Xie. Co-occurrent features in semantic segmentation. In *CVPR*, 2019. 7

[58] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017. 6, 7

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7

[60] H. Zhao, Z. Yi, L. Shu, S. Jianping, C. C. Loy, L. Dahua, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. *ECCV*, 2018. 6, 7

[61] J. Zhao, J. Li, X. Nie, F. Zhao, Y. Chen, Z. Wang, J. Feng, and S. Yan. Self-supervised neural aggregation networks for human parsing. In *CVPRW*, 2017. 7

[62] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 4

[63] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 7

Table 14: Comparison with other refinement mechanisms. LR represents label-refinement and LE represents label-ensemble. All the results are evaluated on Cityscapes val set

Method	Coarse. seg	Fine. seg	mIoU (%)
Baseline	✓	✗	73.90
Baseline	✗	✓	75.80
LE	✓	✓	76.20
LR	✓	✓	77.10
Ours	✓	✓	**79.58**

6. Appendix

Comparison with Refinement. The previous works [14, 24, 12, 44] have exploited various mechanisms to use a set of coarse segmentation maps to boost the final segmentation performance. We mainly compare with two popular approaches including: (i) label-refinement [20, 14]: combine the input image or feature map with a coarse prediction to predict a new refined label map. We concatenate the coarse segmentation maps with the feature map output from after-4 and apply the final classifier on the concatenated feature map to make the final prediction. (ii) label-ensemble [25, 39]: ensemble the coarse predictions with the final predictions directly. We directly use the weighted sum of the coarse segmentation map and the fine segmentation map as the final prediction. Besides, we also report the performance of the coarse segmentation map (prediction from the stage-3 of ResNet) and fine segmentation map (prediction from the stage-4 of ResNet) of the baseline approach (ResNet-101 based FCN). It can be seen that our OCR outperforms all the other approaches by a large margin.