

# Object-Contextual Representations for Semantic Segmentation

Yuhui Yuan<sup>1,2,3</sup>, Xilin Chen<sup>1,2</sup>, and Jingdong Wang<sup>3</sup>

<sup>1</sup> Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS

<sup>2</sup> University of Chinese Academy of Sciences

<sup>3</sup> Microsoft Research Asia

{yuhui.yuan, jingdw}@microsoft.com, xlchen@ict.ac.cn

**Abstract.** In this paper, we study the context aggregation problem in semantic segmentation. Motivated by 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 **learn object regions** under the supervision of the ground-truth segmentation. Second, we **compute the object region representation by aggregating the representations of the pixels lying in the object region**. Last, we compute the relation between each pixel and each object region, and augment the representation of each pixel with the object-contextual representation which is a weighted aggregation of all the object region representations. We empirically demonstrate our method achieves competitive performance on various benchmarks: Cityscapes, ADE20K, LIP, PASCAL-Context and COCO-Stuff. Our submission “HRNet + OCR + SegFix” achieves the 1<sup>st</sup> place on the Cityscapes leaderboard by the ECCV 2020 submission deadline. Code is available at: <https://git.io/openseg> and <https://git.io/HRNet.OCR>.

**Keywords:** Semantic Segmentation; Context Aggregation

## 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. Deep convolutional networks since FCN [47] have been the dominant solutions. Various studies have been conducted, including high-resolution representation learning [7,55], contextual aggregation [79,6] that is the interest of this paper, and so on.

The context of one position typically refers to a set of positions, e.g., the surrounding pixels. The early study is mainly about the spatial scale of contexts, i.e., the spatial scope. Representative works, such as ASPP [6] and PPM [79], exploit multi-scale contexts. Recently, several works, such as DANet [16], CFNet [76]

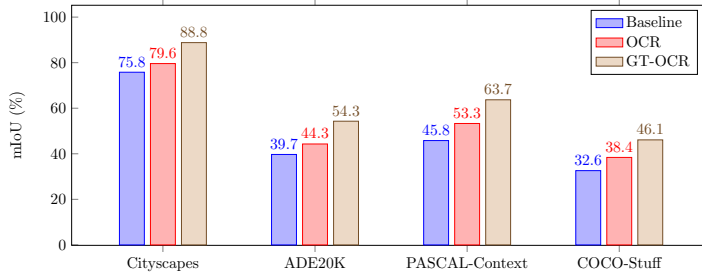


Fig. 1: **Illustrating the effectiveness of our OCR scheme.** GT-OCR estimates the ideal object-contextual representations through exploiting the ground-truth, which is the upper-bound of our method. OCR reports the performance of our proposed object-contextual representations. The three methods, Baseline, OCR and GT-OCR, use the dilated ResNet-101 with output stride 8 as the backbone. We evaluate their (single-scale) segmentation results on Cityscapes **val**, ADE20K **val**, PASCAL-Context **test** and COCO-Stuff **test** separately.

and OCNet [71], consider the relations between a position and its contextual positions, and aggregate the representations of the contextual positions with higher weights for similar representations.

We propose to investigate the contextual representation scheme along the line of exploring the relation between a position and its context. The motivation is that *the class label assigned to one pixel is the category of the object*<sup>4</sup> *that the pixel belongs to*. We aim to augment the representation of one pixel by exploiting the representation of the object region of the corresponding class. The empirical study, shown in Fig. 1, verifies that such a representation augmentation scheme, when the ground-truth object region is given, dramatically improves the segmentation quality<sup>5</sup>.

Our approach consists of three main steps. First, we divide the contextual pixels into a set of soft object regions with each corresponding to a class, i.e., a coarse soft segmentation computed from a deep network (e.g., ResNet [23] or HRNet [55]). Such division is learned under the supervision of the ground-truth segmentation. Second, we estimate the representation for each object region by aggregating the representations of the pixels in the corresponding object region. Last, we augment the representation of each pixel with the object-contextual representation (OCR). The OCR is the weighted aggregation of all the object region representations with the weights calculated according to the relations between pixels and object regions.

The proposed OCR approach differs from the conventional multi-scale context schemes. Our OCR differentiates the same-object-class contextual pixels from the different-object-class contextual pixels, while the multi-scale context

<sup>4</sup> We use “object” to represent both “things” and “stuff” following [14,54].

<sup>5</sup> See Section 3.4 for more details.

schemes, such as ASPP [6] and PPM [79], do not, and only differentiate the pixels with different spatial positions. Fig. 2 provides an example to illustrate the differences between our OCR context and the multi-scale context. On the other hand, our OCR approach is also different from the previous relational context schemes [63,16,71,74,76]. Our approach structures the contextual pixels into object regions and exploits the relations between pixels and object regions. In contrast, the previous relational context schemes consider the contextual pixels separately and only exploit the relations between pixels and contextual pixels [16,71,76] or predict the relations only from pixels without considering the regions [74].

We evaluate our approach on various challenging semantic segmentation benchmarks. Our approach outperforms the multi-scale context schemes, e.g., PSPNet, DeepLabv3, and the recent relational context schemes, e.g., DANet, and the efficiency is also improved. Our approach achieves competitive performance on five benchmarks: 84.5% on Cityscapes **test**, 45.66% on ADE20K **val**, 56.65% on LIP **val**, 56.2% on PASCAL-Context **test** and 40.5% on COCO-Stuff **test**. Besides, we extend our approach to Panoptic-FPN [29] and verify the effectiveness of our OCR on the COCO panoptic segmentation task, e.g., Panoptic-FPN + OCR achieves 44.2% on COCO **val**.

## 2 Related Work

**Multi-scale context.** PSPNet [79] performs regular convolutions on pyramid pooling representations to capture the multi-scale context. The DeepLab series [5,6] adopt parallel dilated convolutions with different dilation rates (each rate captures the context of a different scale). The recent works [22,68,83,71] propose various extensions, e.g., DenseASPP [68] densifies the dilated rates to cover larger scale ranges. Some other studies [7,42,17] construct the encoder-decoder structures to exploit the multi-resolution features as the multi-scale context.

**Relational context.** DANet [16], CFNet [76] and OCNet [71] augment the representation for each pixel by aggregating the representations of the contextual pixels, where the context consists of all the pixels. Different from the global context [46], these works consider the relation (or similarity) between the pixels, which is based on the self-attention scheme [63,61], and perform a weighted aggregation with the similarities as the weights.

Double Attention and its related work [8,74,9,40,38,73,34,24] and ACFNet [74] group the pixels into a set of regions, and then augment the pixel representations by aggregating the region representations with the consideration of their context relations predicted by using the pixel representation.

Our approach is a relational context approach and is related to Double Attention and ACFNet. The differences lie in the region formation and the pixel-region relation computation. Our approach learns the regions with the supervision of the ground-truth segmentation. In contrast, the regions in previous approaches except ACFNet are formed unsupervisedly. On the other hand, the relation between a pixel and a region is computed by considering both the pixel and region



Fig. 2: Illustrating the multi-scale context with the ASPP as an example and the OCR context for the pixel marked with ■. (a) ASPP: The context is a set of sparsely sampled pixels marked with ■, ■. The pixels with different colors correspond to different dilation rates. Those pixels are distributed in both the object region and the background region. (b) Our OCR: The context is expected to be a set of pixels lying in the object (marked with color ■). The image is chosen from ADE20K.

representations, while the relation in previous works is only computed from the pixel representation.

**Coarse-to-fine segmentation.** Various coarse-to-fine segmentation schemes have been developed [15,18,33,59,27,32,84] to gradually refine the segmentation maps from coarse to fine. For example, [33] regards the coarse segmentation map as an additional representation and combines it with the original image or other representations for computing a fine segmentation map.

Our approach in some sense can also be regarded as a coarse-to-fine scheme. The difference lies in that we use the coarse segmentation map for generating a contextual representation instead of directly used as an extra representation. We compare our approach with the conventional coarse-to-fine schemes in the supplementary material.

**Region-wise segmentation.** There exist many region-wise segmentation methods [1,2,21,20,64,50,2,60] that organize the pixels into a set of regions (usually super-pixels), and then classify each region to get the image segmentation result. Our approach does not classify each region for segmentation and instead uses the region to learn a better representation for the pixel, which leads to better pixel labeling.

### 3 Approach

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.

#### 3.1 Background

**Multi-scale context.** The ASPP [5] module captures the multi-scale context information by performing several parallel dilated convolutions with different

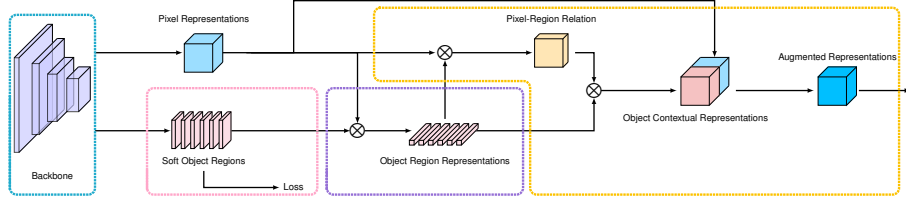


Fig. 3: **Illustrating the pipeline of OCR.** (i) form the soft object regions in the *pink dashed box*. (ii) estimate the object region representations in the *purple dashed box*; (iii) compute the object contextual representations and the augmented representations in the *orange dashed box*. See Section 3.2 and 3.3 for more details.

dilation rates  $[5, 6, 70]$ :

$$\mathbf{y}_i^d = \sum_{\mathbf{p}_s = \mathbf{p}_i + d\Delta_t} \mathbf{K}_t^d \mathbf{x}_s. \quad (1)$$

Here,  $\mathbf{p}_s = \mathbf{p}_i + d\Delta_t$  is the  $s$ th sampled position for the dilation convolution with the dilation rate  $d$  (e.g.,  $d = 12, 24, 36$  in DeepLabv3 [6]) at the position  $\mathbf{p}_i$ .  $t$  is the position index for a convolution, e.g.,  $\{\Delta_t = (\Delta_w, \Delta_h) | \Delta_w = -1, 0, 1, \Delta_h = -1, 0, 1\}$  for a  $3 \times 3$  convolution.  $\mathbf{x}_s$  is the representation at  $\mathbf{p}_s$ .  $\mathbf{y}_i^d$  is the output representation at  $\mathbf{p}_i$  for the  $d$ th dilated convolution.  $\mathbf{K}_t^d$  is the kernel parameter at position  $t$  for the  $d$ th dilated convolution. The output multi-scale contextual representation is the concatenation of the representations output by the parallel dilated convolutions.

The multi-scale context scheme based on dilated convolutions captures the contexts of multiple scales without losing the resolution. The pyramid pooling module in PSPNet [79] performs regular convolutions on representations of different scales, and also captures the contexts of multiple scales but loses the resolution for large scale contexts.

**Relational context.** The relational context scheme [16, 71, 76] computes the context for each pixel by considering the relations:

$$\mathbf{y}_i = \rho\left(\sum_{s \in \mathcal{I}} w_{is} \delta(\mathbf{x}_s)\right), \quad (2)$$

where  $\mathcal{I}$  refers to the set of pixels in the image,  $w_{is}$  is the relation between  $\mathbf{x}_i$  and  $\mathbf{x}_s$ , and may be predicted only from  $\mathbf{x}_i$  or computed from  $\mathbf{x}_i$  and  $\mathbf{x}_s$ .  $\delta(\cdot)$  and  $\rho(\cdot)$  are two different transform functions as done in self-attention [61]. The global context scheme [46] is a special case of relational context with  $w_{is} = \frac{1}{|\mathcal{I}|}$ .

### 3.2 Formulation

The class label  $l_i$  for pixel  $p_i$  is essentially the label of the object that pixel  $p_i$  lies in. Motivated by this, we present an object-contextual representation approach, characterizing each pixel by exploiting the corresponding object representation.

The proposed object-contextual representation scheme (1) structurizes all the pixels in image  $I$  into  $K$  soft object regions, (2) represents each object region as  $\mathbf{f}_k$  by aggregating the representations of all the pixels in the  $k$ th object region, and (3) augments the representation for each pixel by aggregating the  $K$  object region representations with consideration of its relations with all the object regions:

$$\mathbf{y}_i = \rho\left(\sum_{k=1}^K w_{ik} \delta(\mathbf{f}_k)\right), \quad (3)$$

where  $\mathbf{f}_k$  is the representation of the  $k$ th object region,  $w_{ik}$  is the relation between the  $i$ th pixel and the  $k$ th object region.  $\delta(\cdot)$  and  $\rho(\cdot)$  are transformation functions. **Soft object regions.** We partition the image  $I$  into  $K$  soft object regions  $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_K\}$ . Each object region  $\mathbf{M}_k$  corresponds to the class  $k$ , and is represented by a 2D map (or coarse segmentation map), where each entry indicates the degree that the corresponding pixel belongs to the class  $k$ .

We compute the  $K$  object regions from an intermediate representation output from a backbone (e.g., ResNet or HRNet). During training, we learn the object region generator under the supervision from the ground-truth segmentation using the cross-entropy loss.

**Object region representations.** We aggregate the representations of all the pixels weighted by their degrees belonging to the  $k$ th object region, forming the  $k$ th object region representation:

$$\mathbf{f}_k = \sum_{i \in \mathcal{I}} \tilde{m}_{ki} \mathbf{x}_i. \quad (4)$$

Here,  $\mathbf{x}_i$  is the representation of pixel  $p_i$ .  $\tilde{m}_{ki}$  is the normalized degree for pixel  $p_i$  belonging to the  $k$ th object region. We use spatial softmax to normalize each object region  $\mathbf{M}_k$ .

**Object contextual representations.** We compute the relation between each pixel and each object region as below:

$$w_{ik} = \frac{e^{\kappa(\mathbf{x}_i, \mathbf{f}_k)}}{\sum_{j=1}^K e^{\kappa(\mathbf{x}_i, \mathbf{f}_j)}}. \quad (5)$$

Here,  $\kappa(\mathbf{x}, \mathbf{f}) = \phi(\mathbf{x})^\top \psi(\mathbf{f})$  is the unnormalized relation function,  $\phi(\cdot)$  and  $\psi(\cdot)$  are two transformation functions implemented by  $1 \times 1$  conv  $\rightarrow$  BN  $\rightarrow$  ReLU. This is inspired by self-attention [61] for a better relation estimation.

The object contextual representation  $\mathbf{y}_i$  for pixel  $p_i$  is computed according to Equation 3. In this equation,  $\delta(\cdot)$  and  $\rho(\cdot)$  are both transformation functions implemented by  $1 \times 1$  conv  $\rightarrow$  BN  $\rightarrow$  ReLU, and this follows non-local networks [63].

**Augmented representations.** 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  $\mathbf{y}_i$ :

$$\mathbf{z}_i = g([\mathbf{x}_i^\top \ \mathbf{y}_i^\top]^\top). \quad (6)$$

where  $g(\cdot)$  is a transform function used to fuse the original representation and the object contextual representation, implemented by  $1 \times 1$  conv  $\rightarrow$  BN  $\rightarrow$  ReLU. The whole pipeline of our approach is illustrated in Fig. 3.

*Comments:* Some recent studies, e.g., Double Attention [8] and ACFNet [74], can be formulated similarly to Equation 3, but differ from our approach in some aspects. For example, the region formed in Double Attention do not correspond to an object class, and the relation in ACFNet [74] is computed only from the pixel representation w/o using the object region representation.

### 3.3 Architecture

**Backbone.** We use the dilated ResNet-101 [23] (with output stride 8) or HRNet-W48 [55] (with output stride 4) as the backbone. For dilated ResNet-101, there are two representations input to the OCR module. The first representation from Stage 3 is for predicting coarse segmentation (object regions). The other representation from Stage 4 goes through a  $3 \times 3$  convolution (512 output channels), and then is fed into the OCR module. For HRNet-W48, we only use the final representation as the input to the OCR module.

**OCR module.** We implement the above formulation of our approach as the OCR module, as illustrated in Fig. 3. We use a linear function (a  $1 \times 1$  convolution) to predict the coarse segmentation (soft object region) supervised with a pixel-wise cross-entropy loss. All the transform functions,  $\psi(\cdot)$ ,  $\phi(\cdot)$ ,  $\delta(\cdot)$ ,  $\rho(\cdot)$ , and  $g(\cdot)$ , are implemented as  $1 \times 1$  conv  $\rightarrow$  BN  $\rightarrow$  ReLU, and the first three output 256 channels and the last two output 512 channels. We predict the final segmentation from the final representation using a linear function and we also apply a pixel-wise cross-entropy loss on the final segmentation prediction.

### 3.4 Empirical Analysis

We conduct the empirical analysis experiments using the dilated ResNet-101 as the backbone on Cityscapes val.

**Object region supervision.** We study the influence of the object region supervision. We modify our approach through removing the supervision (i.e., loss) on the soft object regions (within the pink dashed box in Fig. 3), and adding another auxiliary loss in the stage-3 of ResNet-101. We keep all the other settings the same and report the results in the left-most 2 columns of Table 1. We can see that the supervision for forming the object regions is crucial for the performance.

**Pixel-region relations.** We compare our approach with other two mechanisms that do not use the region representation for estimating the pixel-region relations: (i) Double-Attention [8] uses the pixel representation to predict the relation; (ii) ACFNet [74] directly uses one intermediate segmentation map to indicate the relations. We use DA scheme and ACF scheme to represent the above two mechanisms. We implement both methods by ourselves and only use the dilated ResNet-101 as the backbone without using multi-scale contexts (the results of ACFNet is improved by using ASPP [74])

Table 1: **Influence of object region supervision and pixel-region relation estimation scheme.** We can find both the object region supervision and our pixel-region relation scheme are important for the performance.

Object region supervision		Pixel-region relations		
w/o supervision	w/ supervision	DA scheme	ACF scheme	Ours
77.31%	<b>79.58%</b>	79.01%	78.02%	<b>79.58%</b>

The comparison in Table 1 shows that our approach gets superior performance. The reason is that we exploit the pixel representation as well as the region representation for computing the relations. The region representation is able to characterize the object in the specific image, and thus the relation is more accurate for the specific image than that only using the pixel representation.

**Ground-truth OCR.** We study the segmentation performance using the ground-truth segmentation to form the object regions and the pixel-region relations, called GT-OCR, to justify our motivation. (i) Object region formation using the ground-truth: set the confidence of pixel  $i$  belonging to  $k$ th object region  $m_{ki} = 1$  if the ground-truth label  $l_i \equiv k$  and  $m_{ki} = 0$  otherwise. (ii) Pixel-region relation computation using the ground-truth: set the pixel-region relation  $w_{ik} = 1$  if the ground-truth label  $l_i \equiv k$  and  $w_{ik} = 0$  otherwise. We have illustrated the detailed results of GT-OCR on four different benchmarks in Fig. 1.

## 4 Experiments: Semantic Segmentation

### 4.1 Datasets

**Cityscapes.** The Cityscapes dataset [11] is tasked for urban scene understanding. There are totally 30 classes and only 19 classes are used for parsing evaluation. The dataset contains 5K high quality pixel-level finely annotated images and 20K coarsely annotated images. The finely annotated 5K images are divided into 2,975/500/1,525 images for training, validation and testing.

**ADE20K.** The ADE20K dataset [81] is used in ImageNet scene parsing challenge 2016. There are 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 dataset [19] is used in the LIP challenge 2016 for single human parsing task. There are about 50K 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-Context.** The PASCAL-Context dataset [49] is a challenging scene parsing dataset that contains 59 semantic classes and 1 background class. The training set and test set consist of 4,998 and 5,105 images respectively.

**COCO-Stuff.** The COCO-Stuff dataset [3] is a challenging scene parsing dataset that contains 171 semantic classes. The training set and test set consist of 9K and 1K images respectively.



## 4.2 Implementation Details

**Training setting.** We initialize the backbones using the model pre-trained on ImageNet and the OCR module randomly. We perform the polynomial learning rate policy with factor  $(1 - (\frac{iter}{iter_{max}})^{0.9})$ , the weight on the final loss as 1, the weight on the loss used to supervise the object region estimation (or auxiliary loss) as 0.4. We use INPLACE-ABN<sup>sync</sup> [53] to synchronize the mean and standard-deviation of BN across multiple GPUs. For the data augmentation, we perform random flipping horizontally, random scaling in the range of  $[0.5, 2]$  and random brightness jittering within the range of  $[-10, 10]$ . We perform the same training settings for the reproduced approaches, e.g., PPM, ASPP, to ensure the fairness. We follow the previous works [6,75,79] for setting up the training for the benchmark datasets.

□ *Cityscapes*: We set the initial learning rate as 0.01, weight decay as 0.0005, crop size as  $769 \times 769$  and batch size as 8 by default. For the experiments evaluated on **val/test** set, we set training iterations as 40K/100K on **train/train+val** set separately. For the experiments augmented with extra data: (i) w/ **coarse**, we first train our model on **train + val** for 100K iterations with initial learning rate as 0.01, then we fine-tune the model on **coarse** set for 50K iterations and continue fine-tune our model on **train+val** for 20K iterations with the same initial learning rate 0.001. (ii) w/ **coarse + Mapillary** [50], we first pre-train our model on the Mapillary **train** set for 500K iterations with batch size 16 and initial learning rate 0.01 (achieves 50.8% on Mapillary **val**), then we fine-tune the model on Cityscapes following the order of **train + val** (100K iterations)  $\rightarrow$  **coarse** (50K iterations)  $\rightarrow$  **train + val** (20K iterations), we set the initial learning rate as 0.001 and the batch size as 8 during the above three fine-tuning stages on Cityscapes.

□ *ADE20K*: We set the initial learning rate as 0.02, weight decay as 0.0001, crop size as  $520 \times 520$ , batch size as 16 and training iterations as 150K if not specified.

□ *LIP*: We set the initial learning rate as 0.007, weight decay as 0.0005, crop size as  $473 \times 473$ , batch size as 32 and training iterations as 100K if not specified.

□ *PASCAL-Context*: We set the initial learning rate as 0.001, weight decay as 0.0001, crop size as  $520 \times 520$ , batch size as 16 and training iterations as 30K if not specified.

□ *COCO-Stuff*: We set the initial learning rate as 0.001, weight decay as 0.0001, crop size as  $520 \times 520$ , batch size as 16 and training iterations as 60K if not specified.

## 4.3 Comparison with Existing Context Schemes

We conduct the experiments using the dilated ResNet-101 as the backbone and use the same training/testing settings to ensure the fairness.

**Multi-scale contexts.** We compare our OCR with the multi-scale context schemes including PPM [79] and ASPP [6] on three benchmarks including Cityscapes

Table 2: **Comparison with multi-scale context scheme.** We use ★ to mark the result w/o using Cityscapes **val** for training. We can find OCR consistently outperforms both PPM and ASPP across different benchmarks under the fair comparisons.

Method	Cityscapes (w/o coarse)	Cityscapes (w/ coarse)	ADE20K	LIP
PPM [79]	78.4%★	81.2%	43.29%	—
ASPP [6]	—	81.3%	—	—
PPM (Our impl.)	80.3%	81.6%	44.50%	54.76%
ASPP (Our impl.)	81.0%	81.7%	44.60%	55.01%
OCR	<b>81.8%</b>	<b>82.4%</b>	<b>45.28%</b>	<b>55.60%</b>

Table 3: **Comparison with relational context scheme.** Our method consistently performs better across different benchmarks. Notably, Double Attention is sensitive to the region number choice and we have fine-tuned this hyper-parameter as 64 to report its best performance.

Method	Cityscapes (w/o coarse)	Cityscapes (w/ coarse)	ADE20K	LIP
CC-Attention [26]	81.4%	—	45.22%	—
DANet [16]	81.5%	—	—	—
Self Attention (Our impl.)	81.1%	82.0%	44.75%	55.15%
Double Attention (Our impl.)	81.2%	82.0%	44.81%	55.12%
OCR	<b>81.8%</b>	<b>82.4%</b>	<b>45.28%</b>	<b>55.60%</b>

Table 4: **Complexity comparison.** We use input feature map of size  $[1 \times 2048 \times 128 \times 128]$  to evaluate their complexity during inference. The numbers are obtained on a single P40 GPU with CUDA 10.0. All the numbers are the smaller the better. Our OCR requires the least GPU memory and the least runtime.

Method	Parameters▲	Memory▲	FLOPs ▲	Time▲
PPM (Our impl.)	23.1M	792M	619G	99ms
ASPP (Our impl.)	15.5M	284M	492G	97ms
DANet (Our impl.)	10.6M	2339M	1110G	121ms
CC-Attention (Our impl.)	10.6M	427M	804G	131ms
Self-Attention (Our impl.)	10.5M	2168M	619G	96ms
Double Attention (Our impl.)	<b>10.2M</b>	209M	<b>338G</b>	46ms
OCR	10.5M	<b>202M</b>	340G	<b>45ms</b>

**test**, ADE20K **val** and LIP **val** in Table 2. Our reproduced PPM/ASPP outperforms the originally reported numbers in [79,6]. From Table 2, it can be seen that our OCR outperforms both multi-scale context schemes by a large margin. For example, the absolute gains of OCR over PPM (ASPP) for the four comparisons are 1.5% (0.8%), 0.8% (0.7%), 0.78% (0.68%), 0.84% (0.5%). To the best of our knowledge, these improvements are already significant considering that the baselines (with dilated ResNet-101) are already strong and the complexity of our OCR is much smaller.

**Relational contexts.** We compare our OCR with various relational context schemes including Self-Attention [61,63], Criss-Cross attention [26] (CC-Attention), DANet [16] and Double Attention [8] on the same three benchmarks including Cityscapes **test**, ADE20K **val** and LIP **val**. For the reproduced Double Attention, we fine-tune the number of the regions (as it is very sensitive to the hyper-parameter choice) and we choose 64 with the best performance. More detailed analysis and comparisons are illustrated in the supplementary material.

According to the results in Table 3, it can be seen that our OCR outperforms these relational context schemes under the fair comparisons. Notably, the complexity of our OCR is much smaller than most of the other methods.

**Complexity.** We compare the efficiency of our OCR with the efficiencies of the multi-scale context schemes and the relational context schemes. We measure the increased parameters, GPU memory, computation complexity (measured by the number of FLOPs) and inference time that are introduced by the context modules, and do not count the complexity from the backbones. The comparison in Table 4 shows the superiority of the proposed OCR scheme.

□ *Parameters:* Most relational context schemes require less parameters compared with the multi-scale context schemes. For example, our OCR only requires 1/2 and 2/3 of the parameters of PPM and ASPP separately.

□ *Memory:* Both our OCR and Double Attention require much less GPU memory compared with the other approaches (e.g., DANet, PPM). For example, our GPU memory consumption is 1/4, 1/10, 1/2, 1/10 of the memory consumption of PPM, DANet, CC-Attention and Self-Attention separately.

□ *FLOPs:* Our OCR only requires 1/2, 7/10, 3/10, 2/5 and 1/2 of the FLOPs based on PPM, ASPP, DANet, CC-Attention and Self-Attention separately.

□ *Running time:* The runtime of OCR is very small: only 1/2, 1/2, 1/3, 1/3 and 1/2 of the runtime with PPM, ASPP, DANet, CC-Attention and Self-Attention separately.

In general, *our OCR is a much better choice if we consider the balance between performance, memory complexity, GFLOPs and running time.*

#### 4.4 Comparison with State-of-the-Art

Considering that different approaches perform improvements on different baselines to achieve the best performance, we categorize the existing works to two groups according to the baselines that they apply: (i) *simple baseline*: dilated ResNet-101 with stride 8; (ii) *advanced baseline*: PSPNet, DeepLabv3, multi-grid (MG), encoder-decoder structures that achieve higher resolution outputs with stride 4 or stronger backbones such as WideResNet-38, Xception-71 and HRNet.

For fair comparison with the two groups fairly, we perform our OCR on a simple baseline (dilated ResNet-101 with stride 8) and an advanced baseline (HRNet-W48 with stride 4). Notably, our improvement with HRNet-W48 (over ResNet-101) is comparable with the gain of the other work based on advanced baseline methods. For example, DGCNet [77] gains 0.7% with Multi-grid while OCR gains 0.6% with stronger backbone on Cityscapes **test**. We summarize all the results in Table 5 and illustrate the comparison details on each benchmark separately as follows.

**Cityscapes.** Compared with the methods based on the simple baseline on Cityscape **test** w/o using the coarse data, our approach achieves the best performance 81.8%, which is already comparable with some methods based on the advanced baselines, e.g. DANet, ACFNet. Our approach achieves better performance 82.4% through exploiting the coarsely annotated images for training.

Table 5: **Comparison with state-of-the-art.** We use M to represent multi-scale context and R to represent relational context. Red, Green, Blue represent the top-3 results. We use †, ‡ and ‡ to mark the result w/o using Cityscapes val, the method using Mapillary dataset and the method using the Cityscapes video dataset separately

Method	Baseline	Stride	Context schemes	Cityscapes (w/o coarse)	Cityscapes (w/ coarse)	ADE20K	LIP	PASCAL Context	COCO-Stuff
Simple baselines									
PSPNet [79]	ResNet-101	8×	M	78.4 <sup>b</sup>	81.2	43.29	-	47.8	-
DeepLabv3 [6]	ResNet-101	8×	M	-	81.3	-	-	-	-
PSANet [80]	ResNet-101	8×	R	80.1	81.4	43.77	-	-	-
SAC [78]	ResNet-101	8×	M	78.1	-	44.30	-	-	-
AAF [28]	ResNet-101	8×	R	79.1 <sup>b</sup>	-	-	-	-	-
DSSPN [41]	ResNet-101	8×	-	77.8	-	43.68	-	-	38.9
DepthSeg [31]	ResNet-101	8×	-	78.2	-	-	-	-	-
MMAN [48]	ResNet-101	8×	-	-	-	-	46.81	-	-
JPPNet [39]	ResNet-101	8×	M	-	-	-	51.37	-	-
EncNet [75]	ResNet-101	8×	-	-	-	44.65	-	51.7	-
GCU [38]	ResNet-101	8×	R	-	-	44.81	-	-	-
APCNet [22]	ResNet-101	8×	M,R	-	-	45.38	-	54.7	-
CFNet [76]	ResNet-101	8×	R	79.6	-	44.89	-	54.0	-
BFP [12]	ResNet-101	8×	R	81.4	-	-	-	53.6	-
CCNet [26]	ResNet-101	8×	R	81.4	-	45.22	-	-	-
ANNet [83]	ResNet-101	8×	M,R	81.3	-	45.24	-	52.8	-
OCR	ResNet-101	8×	R	81.8	82.4	45.28	55.60	54.8	39.5
Advanced baselines									
DenseASPP [68]	DenseNet-161	8×	M	80.6	-	-	-	-	-
DANet [16]	ResNet-101 + MG	8×	R	81.5	-	45.22	-	52.6	39.7
DGCNet [77]	ResNet-101 + MG	8×	R	82.0	-	-	-	53.7	-
EMANet [35]	ResNet-101 + MG	8×	R	-	-	-	-	53.1	39.9
SeENet [52]	ResNet-101 + ASPP	8×	M	81.2	-	-	-	-	-
SGR [40]	ResNet-101 + ASPP	8×	R	-	-	44.32	-	52.5	39.1
OCNet [71]	ResNet-101 + ASPP	8×	M,R	81.7	-	45.45	54.72	-	-
ACFNet [74]	ResNet-101 + ASPP	8×	M,R	81.8	-	-	-	-	-
CNIF [62]	ResNet-101 + ASPP	8×	M	-	-	-	56.93	-	-
GALD [36]	ResNet-101 + ASPP	8×	M,R	81.8	82.9	-	-	-	-
GALD <sup>†</sup> [36]	ResNet-101 + CGNL + MG	8×	M,R	-	83.3	-	-	-	-
Mapillary [53]	WideResNet-38 + ASPP	8×	M	-	82.0	-	-	-	-
GSCNN <sup>‡</sup> [56]	WideResNet-38 + ASPP	8×	M	82.8	-	-	-	-	-
SPGNet [10]	2× ResNet-50	4×	-	81.1	-	-	-	-	-
ZigZagNet [42]	ResNet-101	4×	M	-	-	-	-	52.1	-
SVCNet [13]	ResNet-101	4×	R	81.0	-	-	-	53.2	39.6
ACNet [17]	ResNet-101 + MG	4×	M,R	82.3	-	45.90	-	54.1	40.1
CE2P [45]	ResNet-101 + PPM	4×	M	-	-	-	53.10	-	-
VPLR <sup>†‡</sup> [82]	WideResNet-38 + ASPP	4×	M	-	83.5	-	-	-	-
DeepLabv3+ [7]	Xception-71	4×	M	-	82.1	-	-	-	-
DPC [4]	Xception-71	4×	M	82.7	-	-	-	-	-
DUpsampling [58]	Xception-71	4×	M	-	-	-	-	52.5	-
HRNet [55]	HRNetV2-W48	4×	-	81.6	-	-	55.90	54.0	-
OCR	HRNetV2-W48	4×	R	82.4	83.0	45.66	56.65	56.2	40.5
OCR <sup>†</sup>	HRNetV2-W48	4×	R	83.6	84.2	-	-	-	-

For comparison with the approaches based on the advanced baselines, we perform our OCR on the HRNet-W48, and pre-train our model on the Mapillary dataset [50]. Our approach achieves 84.2% on Cityscapes test. We further apply a novel post-processing scheme SegFix [72] to refine the boundary quality, which brings 0.3% ↑ improvement. Our final submission “HRNet + OCR + SegFix” achieves 84.5%, which ranks the 1<sup>st</sup> place on the Cityscapes leaderboard by the

time of our submission. In fact, we perform PPM and ASPP on HRNet-W48 separately and empirically find that directly applying either PPM or ASPP does not improve the performance and even degrades the performance, while our OCR consistently improves the performance.

Notably, the very recent work [57] sets a new state-of-the-art performance 85.4% on Cityscapes leaderboard via combining our “HRNet + OCR” and a new hierarchical multi-scale attention mechanism.

**ADE20K.** From Table 5, it can be seen that our OCR achieves competitive performance (45.28% and 45.66%) compared with most of the previous approaches based on both simple baselines and advanced baselines. For example, the ACFNet [22] exploits both the multi-scale context and relational context to achieve higher performance. The very recent ACNet [17] achieves the best performance through combining richer local and global contexts.

**LIP.** Our approach achieves the best performance 55.60% on LIP val based on the simple baselines. Applying the stronger backbone HRNetV2-W48 further improves the performance to 56.65%, which outperforms the previous approaches. The very recent work CNIF [62] achieves the best performance (56.93%) through injecting the hierarchical structure knowledge of human parts. Our approach potentially benefit from such hierarchical structural knowledge. All the results are based on only flip testing without multi-scale testing<sup>6</sup>.

**PASCAL-Context.** We evaluate the performance over 59 categories following [55]. It can be seen that our approach outperforms both the previous best methods based on simple baselines and the previous best methods based on advanced baselines. The HRNet-W48 + OCR approach achieves the best performance 56.2%, significantly outperforming the second best, e.g., ACPNet (54.7%) and ACNet (54.1%).

**COCO-Stuff.** It can be seen that our approach achieves the best performance, 39.5% based ResNet-101 and 40.5% based on HRNetV2-48.

**Qualitative Results.** We illustrate the qualitative results in the supplementary material due to the limited pages.

## 5 Experiments: Panoptic Segmentation

To verify the generalization ability of our method, we apply OCR scheme on the more challenging panoptic segmentation task [30], which unifies both the instance segmentation task and the semantic segmentation task.

**Dataset.** We choose the COCO dataset [43] to study the effectiveness of our method on panoptic segmentation. We follow the previous work [29] and uses all 2017 COCO images with 80 thing and 53 stuff classes annotated.

**Training Details.** We follow the default training setup of “COCO Panoptic Segmentation Baselines with Panoptic FPN (3× learning schedule)”<sup>7</sup> in Detectron2 [65]. The reproduced Panoptic FPN reaches higher performance than the

<sup>6</sup> Only few methods adopt multi-scale testing. For example, CNIF [62] gets the improved performance from 56.93% to 57.74%.

<sup>7</sup> [https://github.com/facebookresearch/detectron2/blob/master/MODEL\\_ZOO.md](https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md)

Table 6: **Panoptic segmentation results on COCO val 2017.** The performance of Panoptic-FPN [29] is reproduced based on the official open-source Detectron2 [65] and we use the  $3\times$  learning rate schedule by default. Our OCR consistently improves the PQ performance with both backbones.

Backbone	Method	AP	PQ <sup>Th</sup>	mIoU	PQ <sup>St</sup>	PQ
ResNet-50	Panoptic-FPN	40.0	48.3	42.9	31.2	41.5
	Panoptic-FPN + OCR	40.4 (+0.4)	48.6 (+0.3)	44.3 (+1.4)	33.9 (+2.7)	42.7 (+1.2)
ResNet-101	Panoptic-FPN	42.4	49.7	44.5	32.9	43.0
	Panoptic-FPN + OCR	42.7 (+0.3)	50.2 (+0.5)	45.5 (+1.0)	35.2 (+2.3)	44.2 (+1.2)

original numbers in the paper [29] (Panoptic FPN w/ ResNet-50, PQ: 39.2% / Panoptic FPN w/ ResNet-101, PQ: 40.3%) and we choose the higher reproduced results as our baseline.

In our implementation, we use the original prediction from the semantic segmentation head (within Panoptic-FPN) to compute the soft object regions and then we use a OCR head to predict a refined semantic segmentation map. We set the loss weights on both the original semantic segmentation head and the OCR head as 0.25. All the other training settings are kept the same for fair comparison. We directly use the same OCR implementation (for the semantic segmentation task) without any tuning.

**Results.** In Table 6, we can see that OCR improves the PQ performance of Panoptic-FPN (ResNet-101) from 43.0% to 44.2%, where the main improvements come from better segmentation quality on the *stuff* region measured by mIoU and PQ<sup>St</sup>. Specifically, our OCR improves the mIoU and PQ<sup>St</sup> of Panoptic-FPN (ResNet-101) by 1.0% and 2.3% separately. In general, the performance of “Panoptic-FPN + OCR” is very competitive compared to various recent methods [66,44,69]. We also report the results of Panoptic-FPN with PPM and ASPP to illustrate the advantages of our OCR in the supplementary material.

## 6 Conclusions

In this work, we present an object-contextual representation approach for semantic segmentation. The main reason for the success is that the label of a pixel is the label of the object that the pixel lies in and the pixel representation is strengthened by characterizing each pixel with the corresponding object region representation. We empirically show that our approach brings consistent improvements on various benchmarks.

**Acknowledgement** This work is partially supported by Natural Science Foundation of China under contract No. 61390511, and Frontier Science Key Research Project CAS No. QYZDJ-SSW-JSC009.

## 7 Supplementary

In Section A, we compare our approach to the existing coarse-to-fine approaches. In Section B, we study the influence of the region numbers and illustrate the qualitative results with Double Attention. In Section C, we report the results of Panoptic-FPN + PPM / ASPP on the COCO val 2017 and the results of Panoptic-FPN / Panoptic-FPN + OCR on the COCO test-dev. In Section D, we apply our OCR on MobileNetV2 to verify the effectiveness of our approach for real-time applications. In Section E, we verify the advantage of our OCR over the conventional DeepLabv3 [6] and DeepLabv3+ [7] based on the recent `MMSegmentation` code base [67]. Last, in Section F, we illustrate some examples of the qualitative improvements based on our OCR scheme.

### A. Comparison with Coarse-to-fine Schemes

Many existing studies [15,18,33,59] have exploited various coarse-to-fine schemes to use the coarse segmentation results to boost the final segmentation results. We mainly compare OCR with two popular mechanisms including:

□ label-refinement [18,25]: combine the input image or feature map with a coarse prediction to predict the refined label map. We concatenate the coarse segmentation maps with the feature map output from ResNet-101 Stage 4 and apply the final classifier on the concatenated feature map to predict the refined segmentation maps.

□ label-ensemble [37,51]: ensemble the coarse segmentation maps with the fine segmentation maps directly. We directly use the weighted sum of the coarse segmentation map and the fine segmentation map as the final refined prediction.

Besides, we also report the performance with only the coarse segmentation map (prediction from the ResNet Stage 3) and with only the fine segmentation map (prediction from the ResNet Stage 4). We choose the dilated ResNet-101 as our baseline. According to the results in Table 7, it can be seen that our OCR outperforms all the other coarse-to-fine approaches by a large margin.

### B. Ablation Study of Double Attention

**Number of Regions** We fine-tune the number of regions within Double Attention [8] method and report the results on Cityscapes val in Table 8. We choose  $K=64$  if not specified. Besides, it can be seen that the performance with Double Attention is sensitive to the choice of the number of the regions and our approach (with fixed number of regions) consistently outperforms the Double Attention with different region numbers.

**Qualitative Results** We visualize the predicted regions with Double Attention and the object regions predicted with OCR in Figure 4. It can be seen that the predicted object regions with OCR all correspond to explicit semantic meaning, e.g., road, side-walk and car category separately, while the predicted regions with Double Attention mainly highlight the contour pixels without specific semantic meaning, which might be the main advantages of our approach.

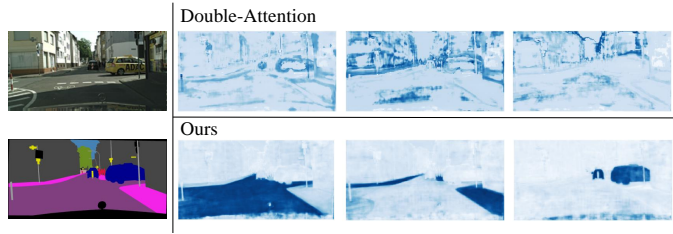


Fig. 4: We randomly choose an image and its ground-truth segmentation map from Cityscapes **val**. The first row illustrates 3 regions predicted with Double Attention and the second row illustrates 3 object regions generated with our OCR. It can be seen that OCR based object regions are more reliable compared to the Double Attention.

Table 7: Comparison with other coarse-to-fine mechanisms. All the results are evaluated on Cityscapes **val**.

Method	Coarse. seg	Fine. seg	mIoU (%)
baseline	✓	✗	73.90
baseline	✗	✓	75.80
label-ensemble	✓	✓	76.20
label-refinement	✓	✓	77.10
OCR	✓	✓	<b>79.58</b>

Table 8: Influence of  $K$  within Double Attention.  $K$  is the number of regions.  $K$  is exact the number of categories for our OCR.

	Double Attention					OCR
# of regions	K=8	K=16	K=32	K=64	K=128	K=19
mIoU	78.52	78.49	78.53	<u>78.65</u>	77.43	<b>79.58</b>

Table 9: **Panoptic segmentation results on COCO val 2017**. The performance of Panoptic-FPN [29] is reproduced based on the official open-source Detectron2 [65] and we use the  $3\times$  learning rate schedule by default. Our OCR consistently outperforms both PPM and ASPP under the fair comparisons.

Backbone	Method	AP	PQ <sup>Th</sup>	mIoU	PQ <sup>St</sup>	PQ
ResNet-50	Panoptic-FPN	40.0	48.3	42.9	31.2	41.5
	Panoptic-FPN + PPM	40.3 (+0.3)	48.3 (+0.0)	43.2 (+0.3)	31.7 (+0.5)	41.7 (+0.2)
	Panoptic-FPN + ASPP	40.2 (+0.2)	48.4 (+0.1)	43.3 (+0.4)	31.8 (+0.6)	41.8 (+0.3)
	Panoptic-FPN + OCR	40.4 (+0.4)	48.6 (+0.3)	44.3 (+1.4)	33.9 (+2.7)	42.7 (+1.2)
ResNet-101	Panoptic-FPN	42.4	49.7	44.5	32.9	43.0
	Panoptic-FPN + PPM	42.5 (+0.1)	50.1 (+0.4)	44.2 (-0.3)	32.8 (-0.1)	43.2 (+0.2)
	Panoptic-FPN + ASPP	42.3 (-0.1)	49.8 (+0.1)	44.4 (-0.1)	33.0 (+0.1)	43.1 (+0.1)
	Panoptic-FPN + OCR	42.7 (+0.3)	50.2 (+0.5)	45.5 (+1.0)	35.2 (+2.3)	44.2 (+1.2)

### C. More Panoptic Segmentation Results

First, we directly apply the PPM or ASPP head before the semantic segmentation head within Panoptic-FPN without any other modifications. In Table 9, we report the results of both methods and we can find our OCR outperforms both the PPM head and the APP head based on Panoptic-FPN. Notably, as illustrated in the paper, our OCR is also more efficient than both PPM and ASPP. Second, we also report the results on the COCO test-dev based on our



Table 10: **Panoptic segmentation results on COCO test-dev.** We submit the results based on Panoptic-FPN / Panoptic-FPN + OCR based on ResNet-101 to the COCO test-dev leaderboard. We also report the original results reported in [29]. Our OCR consistently improves the performance on the COCO test-dev.

Method	PQ <sup>Th</sup>	PQ <sup>St</sup>	PQ
Panoptic-FPN [29]	48.3	29.7	40.9
Panoptic-FPN	50.8	32.4	43.5
Panoptic-FPN + OCR	50.7 ( $-0.1$ )	35.1 ( $+2.7$ )	44.5 ( $+1.0$ )

OCR in Table 10. We can see that our OCR consistently improves the results on both the COCO val set and test-dev set.

#### D. Application to MobileNetV2

We apply the OCR on MobileNetV2 and report the performance in Table 5. Specifically, we train the MobileNetV2 following the same training settings except changing the batch size as 16 and the training iterations as 100K. It can be seen that our OCR significantly improves the segmentation performance on the Cityscapes **val** while slightly increases the inference time (or smaller FPS).

Method	FPS	Cityscapes <b>val</b> mIoU
MobileNetV2	<b>31</b>	69.50%
MobileNetV2 + OCR	28	<b>74.18%</b>

Fig. 5: MobileNetV2 + OCR: Speed (measured by FPS) is tested on P40 GPU with input image of size  $1024 \times 512$

#### E. MMSegmentation Results

To verify that our OCR method generalizes well across different code bases, we further compare the segmentation results of OCR, DeepLabv3 and DeepLabv3+ based on a very recent code base **MMSegmentation** [67]. Specifically speaking, we evaluate different methods under two different training iteration schedules: (i) 40K iterations, (ii) 80K iterations. We set the initial learning rate 0.02 and the batch size 16 for both training schedules. We choose the crop size as  $1024 \times 512$  and the backbone as dilated ResNet-101 with output stride 8 for all methods by default to ensure the fairness of the comparison.

We report the GPU memory consumption (for training), inference speed (for testing) and mIoUs (on Cityscapes validation set) in Table 11. We can see that OCR achieves better or comparable performance compared to DeepLabv3 and DeepLabv3+ under both kinds of training settings. Especially, our OCR requires less GPU memory consumption and achieve higher FPS on Cityscapes benchmark.

Table 11: **Cityscapes results based on MMSegmentation code base.** The GPU memory consumption is the smaller the better and both FPS and mIoU are the larger the better. The mIoU is evaluated on Cityscapes validation set w/o using flip and multi-scale testing. The GPU Memory consumption is tested with 2 images on each GPU during training. The FPS is tested based on processing 1 image with resolution  $1024 \times 2048$  on a single GPU. We all use Tesla V100 GPU and Pytorch 1.5.1 for experiments.

Method	iterations	GPU Mem (GB)	FPS	mIoU
DeepLabv3	40K	9.6	2	79.69
DeepLabv3	80K			80.43
DeepLabv3+	40K	11	2.64	80.13
DeepLabv3+	80K			<b>80.86</b>
OCR	40K	<b>8.8</b>	<b>3.02</b>	80.30
OCR	80K			80.81

## F. Qualitative Improvements

We illustrate the qualitative improvements of our method in Fig. 6 on different benchmarks. We use white dashed boxes to mark the hard regions that are well-classified by our approach but mis-classified by the baseline.

## References

1. Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: CVPR (2012)
2. Caesar, H., Uijlings, J., Ferrari, V.: Region-based semantic segmentation with end-to-end training. In: ECCV (2016)
3. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
4. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: NIPS (2018)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. PAMI (2018)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
8. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: A<sup>2</sup>-nets: Double attention networks. In: NIPS (2018)
9. Chen, Y., Rohrbach, M., Yan, Z., Yan, S., Feng, J., Kalantidis, Y.: Graph-based global reasoning networks. arXiv:1811.12814 (2018)

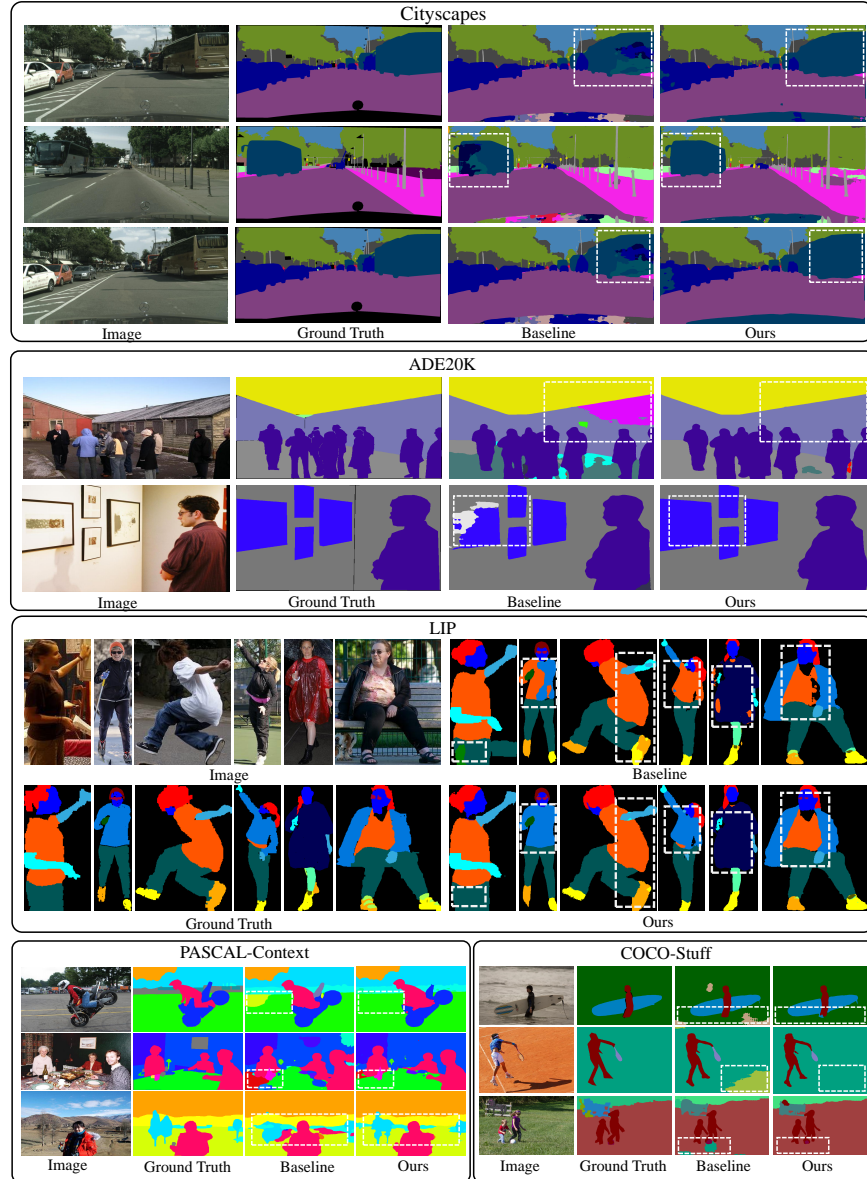


Fig. 6: **Qualitative comparisons.** We compare the segmentation results with dilated ResNet-101 (baseline) and dilated ResNet-101 + OCR (ours) on the 5 benchmarks. We mark the improved regions with white dashed boxes.

10. Cheng, B., Chen, L.C., Wei, Y., Zhu, Y., Huang, Z., Xiong, J., Huang, T.S., Hwu, W.M., Shi, H.: Spynet: Semantic prediction guidance for scene parsing. In: ICCV

- (2019)
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
  12. Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. ICCV (2019)
  13. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Semantic correlation promoted shape-variant context for segmentation. In: CVPR (2019)
  14. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. PAMI (2012)
  15. Fieraru, M., Khoreva, A., Pishchulin, L., Schiele, B.: Learning to refine human pose estimation. In: CVPRW (2018)
  16. Fu, J., Liu, J., Tian, H., Fang, Z., Lu, H.: Dual attention network for scene segmentation. arXiv:1809.02983 (2018)
  17. Fu, J., Liu, J., Wang, Y., Li, Y., Bao, Y., Tang, J., Lu, H.: Adaptive context network for scene parsing. In: ICCV (2019)
  18. Gidaris, S., Komodakis, N.: Detect, replace, refine: Deep structured prediction for pixel wise labeling. In: CVPR (2017)
  19. Gong, K., Liang, X., Zhang, D., Shen, X., Lin, L.: Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In: CVPR (2017)
  20. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
  21. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
  22. He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: CVPR (2019)
  23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
  24. Huang, L., Yuan, Y., Guo, J., Zhang, C., Chen, X., Wang, J.: Interlaced sparse self-attention for semantic segmentation. arXiv preprint arXiv:1907.12273 (2019)
  25. Huang, Y.H., Jia, X., Georgoulis, S., Tuytelaars, T., Van Gool, L.: Error correction for dense semantic image labeling. In: CVPRW (2018)
  26. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: ICCV (2019)
  27. Islam, M.A., Naha, S., Roohan, M., Bruce, N., Wang, Y.: Label refinement network for coarse-to-fine semantic segmentation. arXiv:1703.00551 (2017)
  28. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: ECCV (2018)
  29. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019)
  30. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR (2019)
  31. Kong, S., Fowlkes, C.C.: Recurrent scene parsing with perspective understanding in the loop. In: CVPR (2018)
  32. Kuo, W., Angelova, A., Malik, J., Lin, T.Y.: Shapemask: Learning to segment novel objects by refining shape priors (2019)
  33. Li, K., Hariharan, B., Malik, J.: Iterative instance segmentation. In: CVPR (2016)
  34. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: ICCV (2019)

35. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: ICCV (2019)
36. Li, X., Zhang, L., You, A., Yang, M., Yang, K., Tong, Y.: Global aggregation then local distribution in fully convolutional networks. BMVC (2019)
37. Li, X., Liu, Z., Luo, P., Change Loy, C., Tang, X.: Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In: CVPR (2017)
38. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: NIPS (2018)
39. Liang, X., Gong, K., Shen, X., Lin, L.: Look into person: Joint body parsing & pose estimation network and a new benchmark. PAMI (2018)
40. Liang, X., Hu, Z., Zhang, H., Lin, L., Xing, E.P.: Symbolic graph reasoning meets convolutions. In: NIPS (2018)
41. Liang, X., Zhou, H., Xing, E.: Dynamic-structured semantic propagation network. In: CVPR (2018)
42. Lin, D., Shen, D., Shen, S., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Zigzag-net: Fusing top-down and bottom-up context for object segmentation. In: CVPR (2019)
43. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
44. Liu, H., Peng, C., Yu, C., Wang, J., Liu, X., Yu, G., Jiang, W.: An end-to-end network for panoptic segmentation. In: CVPR (2019)
45. Liu, T., Ruan, T., Huang, Z., Wei, Y., Wei, S., Zhao, Y., Huang, T.: Devil in the details: Towards accurate single and multiple human parsing. arXiv:1809.05996 (2018)
46. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv:1506.04579 (2015)
47. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
48. Luo, Y., Zheng, Z., Zheng, L., Tao, G., Junqing, Y., Yang, Y.: Macro-micro adversarial network for human parsing. In: ECCV (2018)
49. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
50. Neuhold, G., Ollmann, T., Rota Bulo, S., Kotschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: CVPR (2017)
51. Nigam, I., Huang, C., Ramanan, D.: Ensemble knowledge transfer for semantic segmentation. In: WACV (2018)
52. Pang, Y., Li, Y., Shen, J., Shao, L.: Towards bridging semantic gap to improve semantic segmentation. In: ICCV (2019)
53. Rota Bul, S., Porzi, L., Kotschieder, P.: In-place activated batchnorm for memory-optimized training of dnns. In: CVPR (2018)
54. Shetty, R., Schiele, B., Fritz, M.: Not using the car to see the sidewalk—quantifying and controlling the effects of context in classification and segmentation. In: CVPR (2019)
55. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv:1904.04514 (2019)
56. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. ICCV (2019)
57. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation. arXiv:2005.10821 (2020)

58. Tian, Z., He, T., Shen, C., Yan, Y.: Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In: CVPR (2019)
59. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. PAMI (2010)
60. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV (2013)
61. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
62. Wang, W., Zhang, Z., Qi, S., Shen, J., Pang, Y., Shao, L.: Learning compositional neural information fusion for human parsing. In: ICCV (2019)
63. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
64. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR (2017)
65. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. <https://github.com/facebookresearch/detectron2> (2019)
66. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: CVPR (2019)
67. Xu, J., Chen, K., Lin, D.: MMSegmentation. <https://github.com/open-mmlab/mms Segmentation> (2020)
68. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: CVPR (2018)
69. Yang, Y., Li, H., Li, X., Zhao, Q., Wu, J., Lin, Z.: Sognet: Scene overlap graph network for panoptic segmentation. arXiv:1911.07527 (2019)
70. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. ICLR (2016)
71. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv:1809.00916 (2018)
72. Yuan, Y., Xie, J., Chen, X., Wang, J.: Segfix: Model-agnostic boundary refinement for segmentation. In: ECCV (2020)
73. Yue, K., Sun, M., Yuan, Y., Zhou, F., Ding, E., Xu, F.: Compact generalized non-local network. In: NIPS (2018)
74. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: Acfnnet: Attentional class feature network for semantic segmentation. In: ICCV (2019)
75. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: CVPR (2018)
76. Zhang, H., Zhang, H., Wang, C., Xie, J.: Co-occurrent features in semantic segmentation. In: CVPR (2019)
77. Zhang, L., Li, X., Arnab, A., Yang, K., Tong, Y., Torr, P.H.: Dual graph convolutional network for semantic segmentation. BMVC (2019)
78. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: ICCV (2017)
79. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
80. Zhao, H., Yi, Z., Shu, L., Jianping, S., Loy, C.C., Dahua, L., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. ECCV (2018)
81. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017)

- 82. Zhu, Y., Sapra, K., Reda, F.A., Shih, K.J., Newsam, S., Tao, A., Catanzaro, B.: Improving semantic segmentation via video propagation and label relaxation. In: CVPR (2019)
- 83. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: ICCV (2019)
- 84. Zhu, Z., Xia, Y., Shen, W., Fishman, E., Yuille, A.: A 3d coarse-to-fine framework for volumetric medical image segmentation. In: 3DV (2018)