1. Objectness Region Enhancement Networks for Scene Parsing

Accession number: 20172903958928

Authors: Ou, Xin-Yu (1, 2, 3); Li, Ping (1); Ling, He-Fei (1); Liu, Si (2); Wang, Tian-Jiang (1); Li, Dan (1)
Author affiliation: (1) School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan; 430074, China; (2) Institute of Information Engineering, Chinese Academy of Sciences, Beijing; 100091, China; (3) Cadres Online Learning Institute of Yunnan Province, Yunnan Open University, Kunming; 650223, China
Corresponding author: Li, Ping(lpshome@hust.edu.cn)

Source title: Journal of Computer Science and Technology
Abbreviated source title: J Comput Sci Technol
Volume: 32
Issue: 4
Issue date: July 1, 2017
Publication year: 2017
Pages: 683-700
Language: English
ISSN: 10009000
E-ISSN: 18604749
CODEN: JCTEEM
Document type: Journal article (JA)
Publisher: Springer New York LLC

Abstract: Semantic segmentation has recently witnessed rapid progress, but existing methods only focus on identifying objects or instances. In this work, we aim to address the task of semantic understanding of scenes with deep learning. Different from many existing methods, our method focuses on putting forward some techniques to improve the existing algorithms, rather than to propose a whole new framework. Objectness enhancement is the first effective technique. It exploits the detection module to produce object region proposals with category probability, and these regions are used to weight the parsing feature map directly. “Extra background” category, as a specific category, is often attached to the category space for improving parsing result in semantic and instance segmentation tasks. In scene parsing tasks, extra background category is still beneficial to improve the model in training. However, some pixels may be assigned into this nonexistent category in inference. Black-hole filling technique is proposed to avoid the incorrect classification. For verifying these two techniques, we integrate them into a parsing framework for generating parsing result. We call this unified framework as Objectness Enhancement Network (OENet). Compared with previous work, our proposed OENet system effectively improves the performance over the original model on SceneParse150 scene parsing dataset, reaching 38.4 mIoU (mean intersectionover-union) and 77.9% accuracy in the validation set without assembling multiple models. Its effectiveness is also verified on the Cityscapes dataset. © 2017, Springer Science+Business Media, LLC.

Number of references: 40
Main heading: Object detection
Controlled terms: Deep learning - Semantics
Uncontrolled terms: Black holes - Detection modules - objectness region proposal - Original model - scene parsing - Semantic segmentation - Semantic understanding - Unified framework
Classification code: 723.2 Data Processing and Image Processing
Numerical data indexing: Percentage 7.79e+01%
DOI: 10.1007/s11390-017-1751-x
Compendex references: YES
Database: Compendex

Data Provider: Engineering Village

关闭 **Web of Science** 打印

第 **1** 页 **(**记录 **1 -- 1)**

[1]

第 **1** 条，共 **1** 条

标题**:** Objectness Region Enhancement Networks for Scene Parsing

作者**:** Ou, XY (Ou, Xin-Yu); Li, P (Li, Ping); Ling, HF (Ling, He-Fei); Liu, S (Liu, Si); Wang, TJ (Wang, Tian-Jiang); Li, D (Li, Dan)

来源出版物**:** JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 卷**:** 32 期**:** 4 页**:** 683-700 **DOI:** 10.1007/s11390-017-1751-x 出版年**:** JUL 2017

Web of Science 核心合集中的 "被引频次"**:** 0
被引频次合计**:** 0
使用次数 **(**最近 **180** 天**):** 2
使用次数 **(2013** 年至今**):** 2
引用的参考文献数**:** 40

摘要**:** Semantic segmentation has recently witnessed rapid progress, but existing methods only focus on identifying objects or instances. In this work, we aim to address the task of semantic understanding of scenes with deep learning. Different from many existing methods, our method focuses on putting forward some techniques to improve the existing algorithms, rather than to propose a whole new framework. Objectness enhancement is the first effective technique. It exploits the detection module to produce object region proposals with category probability, and these regions are used to weight the parsing feature map directly. "Extra background" category, as a specific category, is often attached to the category space for improving parsing result in semantic and instance segmentation tasks. In scene parsing tasks, extra background category is still beneficial to improve the model in training. However, some pixels may be assigned into this nonexistent category in inference. Black-hole filling technique is proposed to avoid the incorrect classification. For verifying these two techniques, we integrate them into a parsing framework for generating parsing result. We call this unified framework as Objectness Enhancement Network (OENet). Compared with previous work, our proposed OENet system effectively improves the performance over the original model on SceneParse150 scene parsing dataset, reaching 38.4 mIoU (mean intersection-over-union) and 77.9% accuracy in the validation set without assembling multiple models. Its effectiveness is also verified on the Cityscapes dataset.

入藏号**:** WOS:000405580700003
语种**:** English
文献类型**:** Article

作者关键词**:** objectness region enhancement; black-hole filling; scene parsing; instance enhancement; objectness region proposal

地址**:** [Ou, Xin-Yu; Li, Ping; Ling, He-Fei; Wang, Tian-Jiang; Li, Dan] Huazhong Univ Sci & Technol, Sch Comp Sci & Technol, Wuhan 430074, Peoples R China.
[Ou, Xin-Yu; Liu, Si] Chinese Acad Sci, Inst Informat Engn, Beijing 100091, Peoples R China.
[Ou, Xin-Yu] Yunnan Open Univ, Cadres Online Learning Inst Yunnan Prov, Kunming 650223, Peoples R China.
通讯作者地址**:** Li, P (通讯作者),Huazhong Univ Sci & Technol, Sch Comp Sci & Technol, Wuhan 430074, Peoples R China.
电子邮件地址**:** ouxinyu@hust.edu.cn; lpshome@hust.edu.cn; lhefei@hust.edu.cn; liusi@iie.ac.cn; tjwang@hust.edu.cn; lidanhus@hust.edu.cn

出版商**:** SCIENCE PRESS
出版商地址**:** 16 DONGHUANGCHENGGEN NORTH ST, BEIJING 100717, PEOPLES R CHINA

Web of Science 类别**:** Computer Science, Hardware & Architecture; Computer Science, Software Engineering
研究方向**:** Computer Science
IDS 号**:** FA6UV
ISSN: 1000-9000
eISSN: 1860-4749
29 字符的来源出版物名称缩写**:** J COMPUT SCI TECH-CH
ISO 来源出版物缩写**:** J. Comput. Sci. Technol.
来源出版物页码计数**:** 18

基金资助致谢:

基金资助机构	授权号
National Natural Science Foundation of China	U1536203 61572493 61572214 61502185
Major Scientific and Technological Innovation Project of Hubei Province of China	2015AAA013
Open Project Program of the National Laboratory of Pattern Recognition of China	201600035
Key Program of the Natural Science Foundation of the Open University of China	G16F3702Z
Young Scientists Fund of the Natural Science Foundation of the Open University of China	G16F2505Q

This work was supported by the Joint Funds of the National Natural Science Foundation of China under Grant No. U1536203, the National Natural Science Foundation of China under Grant Nos. 61572493, 61572214, and 61502185, the Major Scientific and Technological Innovation Project of Hubei Province of China under Grant No. 2015AAA013, the Open Project Program of the National Laboratory of Pattern Recognition of China under Grant No. 201600035, the Key Program of the Natural Science Foundation of the Open University of China under Grant No. G16F3702Z, and the Young Scientists Fund of the Natural Science Foundation of the Open University of China under Grant No. G16F2505Q.

公开访问**:** No
输出日期**:** 2017-10-24

关闭 **Web of Science** 打印

第 **1** 页 **(**记录 **1 -- 1)**

[1]

Ou XY, Li P, Ling HF *et al.* Objectness region enhancement networks for scene parsing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(4): 683–701 July 2017. DOI 10.1007/s11390-017-1751-x

Objectness Region Enhancement Networks for Scene Parsing

Xin-Yu Ou[1,2,3], *Member, CCF, IEEE*, Ping Li[1,*], He-Fei Ling[1], *Member, CCF, ACM, IEEE*
Si Liu[2], *Member, CCF, ACM, IEEE*, Tian-Jiang Wang[1], *Member, CCF, ACM, IEEE*, and Dan Li[1]

[1] *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*
[2] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100091, China*
[3] *Cadres Online Learning Institute of Yunnan Province, Yunnan Open University, Kunming 650223, China*

E-mail: {ouxinyu, lpshome, lhefei}@hust.edu.cn; liusi@iie.ac.cn; {tjwang, lidanhust}@hust.edu.cn

Received December 20, 2016; revised June 12, 2017.

Abstract Semantic segmentation has recently witnessed rapid progress, but existing methods only focus on identifying objects or instances. In this work, we aim to address the task of semantic understanding of scenes with deep learning. Different from many existing methods, our method focuses on putting forward some techniques to improve the existing algorithms, rather than to propose a whole new framework. Objectness enhancement is the first effective technique. It exploits the detection module to produce object region proposals with category probability, and these regions are used to weight the parsing feature map directly. "Extra background" category, as a specific category, is often attached to the category space for improving parsing result in semantic and instance segmentation tasks. In scene parsing tasks, extra background category is still beneficial to improve the model in training. However, some pixels may be assigned into this nonexistent category in inference. Black-hole filling technique is proposed to avoid the incorrect classification. For verifying these two techniques, we integrate them into a parsing framework for generating parsing result. We call this unified framework as Objectness Enhancement Network (OENet). Compared with previous work, our proposed OENet system effectively improves the performance over the original model on SceneParse150 scene parsing dataset, reaching 38.4 mIoU (mean intersection-over-union) and 77.9% accuracy in the validation set without assembling multiple models. Its effectiveness is also verified in the Cityscapes dataset.

Keywords objectness region enhancement, black-hole filling, scene parsing, instance enhancement, objectness region proposal

1 Introduction

Scene parsing[1-2], or recognizing and segmenting objects and stuffs in an image, is one of the key problems in scene understanding. As an important computer vision task, it can affect every aspect of our lives, such as content-aware search[3-5], scene understanding, autopilot[6], robot navigation[4] and so on.

Nowadays, given a visual scene of a dining room, a service robot equipped for providing services to customers can accurately recognize the scene category and locate its own coordinates. However, to freely navigate in the scene and manipulate the objects inside, the robot needs far more information to comprehend. It needs to recognize and localize not only the notable objects like a table, chair and person, but also small objects like a dish, pepper pot or candy box, and their parts like the handle of a cup or the surface of a table, to allow a potential interaction. It is also very important

Regular Paper

Special Issue on Deep Learning

This work was supported by the Joint Funds of the National Natural Science Foundation of China under Grant No. U1536203, the National Natural Science Foundation of China under Grant Nos. 61572493, 61572214, and 61502185, the Major Scientific and Technological Innovation Project of Hubei Province of China under Grant No. 2015AAA013, the Open Project Program of the National Laboratory of Pattern Recognition of China under Grant No. 201600035, the Key Program of the Natural Science Foundation of the Open University of China under Grant No. G16F3702Z, and the Young Scientists Fund of the Natural Science Foundation of the Open University of China under Grant No. G16F2505Q.

[*] Corresponding Author

for the robot to identify the stuffs like a wall, floor, and door for spatial navigation. Recently, tremendous progresses in semantic segmentation have been made based on the framework of fully convolutional neural networks (FCN)[7]. By reusing the computed feature maps for an image, FCN avoids redundant re-computation for classifying individual pixels in the image. FCN becomes the de facto approach for dense prediction, and many methods were proposed for further improving this framework, such as DeepLab[1] and Adelaide-Context model[8].

However, the pixel-wise prediction in FCN[7] is achieved by roughing up sampling convolutional feature maps via large-span bilinear interpolation. Hence, the boundaries of objects are oversmoothed in the segmentation, and the fixed-size receptive fields possibly make foreground objects overwhelmed by a large area of diverse backgrounds and stuffs, especially those of smaller objects. For semantic and instance segmentation tasks, which concentrate on separating the objects from their background, this is not a big problem. Even in the complex MSCOCO[9] dataset, most objects can be easily found and identified. This is primarily because the scale of an object is usually large enough to find the object easily in the object-based segmentation task. Meanwhile, we do not have to concern about what the background is. Based on these two points, the difficulty of the segmentation task will be reduced. However, in the scene parsing task, complex scene makes most objects very small, and the number and the type of the objects are big and various respectively. Moreover, scene parsing not only segments objects from the scene, but also needs to identify what the backgrounds and stuffs are. Therefore, we need a way to find out the objects from the scene, especially those smaller or ambiguous objects. Several researchers[10-11] proposed using detection to help object segmentation. These methods first use detection to generate region proposals, and then run segmentation in these regions. Detection-based methods are beneficial to recall some missing objects. These objects are difficult to identify in the original segmentation network. However, in scene parsing task, the segmentation in region proposals can make some background pixels incorrectly identified as an object. Moreover, it still needs an extra network to deal with the backgrounds and stuffs, because the region proposals cannot cover all the pixels, and they do not care what the backgrounds and stuffs are. In contrast, we do not parse the scene in the region proposals, but use the region proposals to enhance local features over parsing results. Specifically, we only weight the specific feature channel, which is equal to the index of the predicted category corresponding to an object. Furthermore, we utilize the internal area of the object contour as the object mask to replace the enclosing rectangle to achieve enhancement. This strategy avoids the objectness enhancement being applied in the regions of stuffs or backgrounds. We think only weighting the specified feature channel related to the target object can minimize the number of false matches. Even if the background area of the specified feature channel is weighted by some algorithm, the probabilities of the background pixels are not very high. The main reason is that the initial output probabilities of these pixels are very small. The highest probability of these areas will appear in the feature channel, which is closer to the real category. In order to understand this idea better, we visualize this method in Fig.1.

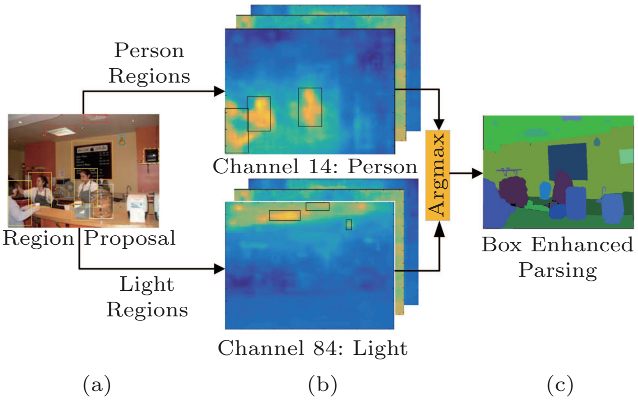

Fig.1. Objectness region enhancement. (a) Output of the OPN (objectness proposal network), which produces region proposals with category information. (b) Region enhancement over different feature map channels. (c) Final parsing result. Note that the region enhancement happens only when the index of the feature map is equal to the category index of the region proposal.

On the other hand, in both the detection and the segmentation tasks, some regions are hard to be determined what they are. Many algorithms[1,7,12-14] add an extra background category[①] to collect the negative samples or marginal samples in training. This policy helps to train a better model, but it leads to that some pixels are assigned to the extra background classes in inference. This is not a problem for semantic and instance segmentation tasks, and at least it is

[①] We define the category which is used for improving the model in training as "extra background" category, and define the categories (such as the sky, ground, wall and grass) which are existing in the original category space as "background" categories.

not obvious in the visual view. Because semantic segmentation and instance segmentation focus on recognizing the specified categories, all the other pixels can be considered as "extra background". In other words, the "extra background" category is a real existing category, and all non-target areas will be identified as extra background. In contrast, we must address each pixel and assign a category to them in scene parsing. To add an extra background category in training, some pixels may be considered as the extra background in the inference, even using CRF (conditional random field) to optimize the parsing results. Normally, the "extra background" category is usually encoded as "0". Under this setting, the areas being assigned to the "extra background" look like black holes in the visual view. Therefore, we call this phenomenon as "black-hole". To tackle this problem, we use the category which has the second high classification probability to replace the extra background category. We think this category may be closer to the true category. We call this simple algorithm as "black-hold filling".

In order to achieve these two strategies for scene parsing, we build an unified framework based on the Deeplab[1] model. Our model consists of three subnetworks as shown in Fig.2. The first one is a feature extraction network (FEN) used to produce convolutional feature maps. The second one is an objectness proposal network (OPN) used to locate and recognize objects in the image. And the last one is an objectness enhancement network (OEN) used to optimize the pixel-level prediction for further semantic segmentation. With this framework, we boost the performance of parsing with detection technique and black-hole filling strategy. Specifically, the detection technique is employed for box-level instance enhancement and mask-level instance enhancement. After instance enhancement, the fully-connected CRF technique is utilized to refine localized region and recover object boundaries. Finally, a black-hole filling strategy is used to deal with the problem of allocating objects/stuffs as "extra background".

Our main contributions can be summarized in three aspects.

1) We propose a unified framework to address the task of scene parsing. Benefiting from the modular design, our improved algorithm can be considered as a series of post-processing methods, and the basic CNN component can be easily replaced by other contemporary deep models for improving overall performance.

2) We propose an objectness enhancement method to recall the ignored objects that are hard to be recognized in the standard scene parsing networks.

3) A "black-hole" filling technique is designed to handle the problem of those pixels beyond the category space.

2 Related Work

2.1 Semantic Scene Parsing

With the success of convolutional neural network for image classification[15], there is grow-

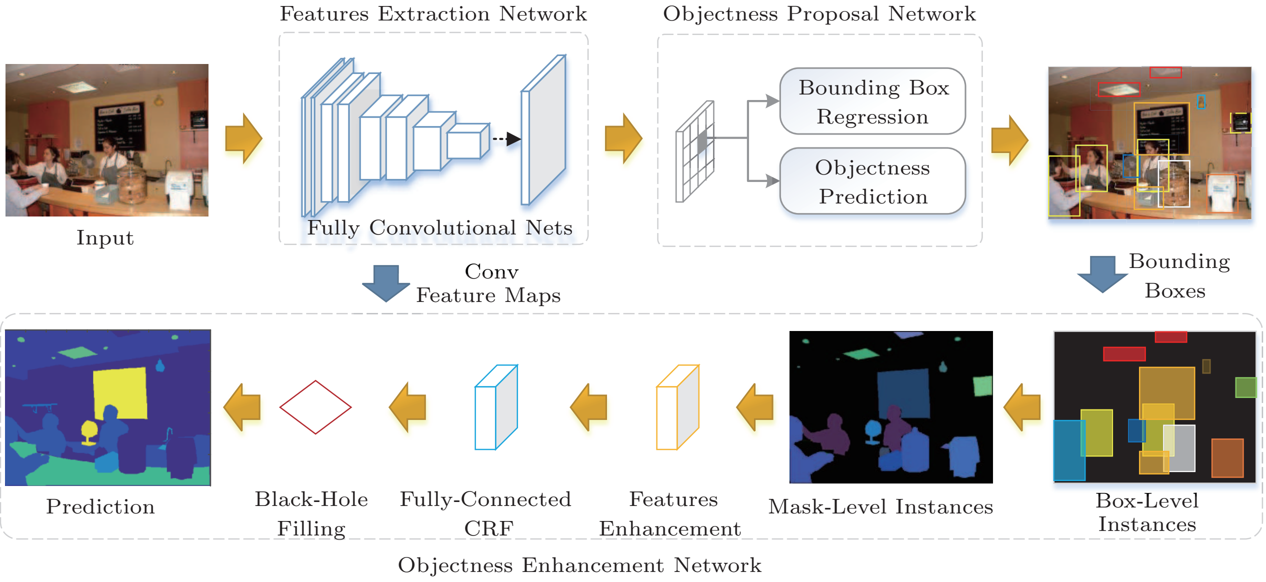

Fig.2. Overall architecture of the OENet framework.

ing interest for semantic pixel-wise labeling using CNN with dense output, such as the fully convolutional network[7], DilatedNet[1], multi-task cascade network[10], encoder-decoder SegNet[16], and deconvolutional neural networks[17]. Most of these methods are corresponding to general object parsing[1,18] and human parsing[19-23]. Instead of the segment object from scenes on pixel-wise, we solve the scene parsing problem which needs to capture and identify both the objects and the stuffs. This is an exceedingly challenging problem, especially when the scene includes plenty of objects in complicated environments. Zhou *et al.*[2,24] did many studies in this field, and they released two datasets: SceneParse150[2] and Places[24]. Many objects semantic segmentation methods are used to handle the scene parsing task. However, in pixel-wise prediction tasks, some of the smaller objects are possibly overwhelmed in the diverse background areas due to the gradually pooled receptive fields. In order to solve this problem, Chen *et al.*[1] developed a powerful "atrous convolution" to explicitly control the resolution and effectively enlarge the field of view of filters to incorporate larger context. Lin *et al.*[8] explored "patch-patch" context and "patch-background" context in deep CNN to improve semantic segmentation.

2.2 Region Representations and Region Proposals

Most traditional studies use hand-crafted features for region-based representation. More recent studies instead use the last convolutional feature map of CNN[25] as feature representations[26-28]. These representations can free-form represent the shape of a region[26-30] or simply represent the bounding box around the region[26]. Furthermore, regions can be cropped out from the image before being fed to the network[27-28] or one can create region representations from a convolutional layer[26,29-30], termed region-of-interest pooling[12,31] or convolutional feature masking[26].

CNN representations become more powerful when further trained for target tasks. Adopting region proposal methods for producing multiple instance proposals, RCNN[27] achieved a milestone performance. Following this work, many pioneering object detection methods[12-14,27,31] were proposed to predict bounding boxes and categories simultaneously, which are implemented with CNN-based deep learning architecture. In SPPnet[31] and Fast RCNN[12], the convolutional layers of CNN are shared on the entire image for fast computation. Faster RCNN[13-14] further exploits the shared convolutional features to extract region proposals.

The output of the detection system usually is a set of bounding boxes that contain a lot of background pixels. Using the system to help segmentation may cause erroneous judgement in overlapping areas. Mask-level instance region proposals can be addressed based on the detection philosophy, such as R-CNN[12-14], SDS[28]. Mask layer[26] is often used to share convolutional features among mask-level proposals. Many methods[11,26-28] rely on computationally expensive mask proposal methods. DeepMask[11] learns segmentation candidates in one second, but its accuracy is yet to be assessed. Category-wise semantic segmentation FCN[7] enables per-pixel regression in a fully-convolution form, but cannot distinguish instances from the same category. [32] improves FCN, but is also powerless to distinguish instances. Dai *et al.*[10] proposed a multi-task network to generate mask-level region proposals and segment objects, but the network does not handle the stuff that is an important part of scene parsing. [33] demonstrates that, as a higher order potential, object detections can be included in a CRF embedded within a deep network. By an end-to-end trainable incorporated CRF, the energy formulation can reject erroneous detections. However, due to the integrated CRF and detection network, it is difficult to improve performance through replacing a better detector. In this paper, we use mask-level instance enhancement to reduce the recognition errors. The modular design methodology allows to improve the overall performance by upgrading components.

In this paper, we also share convolutional features to speed up producing features and proposals for our scene parsing systems, like [12-13, 31]. Some region proposal methods[13-14] are used to generate box-level and mask-level instances. Different from these region-based semantic segmentation methods, our method makes use of the category-wise region proposal to enhance the pixel-wise prediction. Category-wise convolutional feature map and category-wise region proposal are assembled to form a more robust pixel-level feature map, which compensates the weak response objectness in vanilla semantic segmentation task.

3 Methods

With the development of deep learning, using CNN for semantic segmentation has been shown to

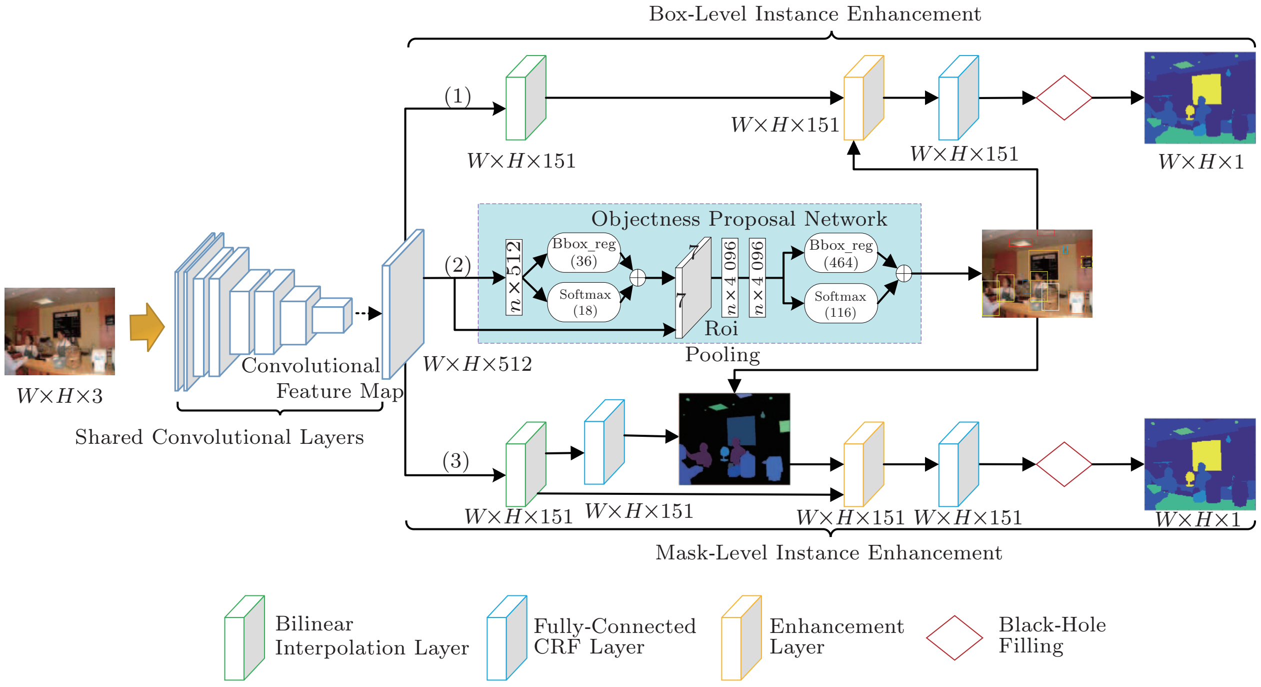

Fig.3. Detailed architecture of the proposed OENet.

be easy and successfully dealt in fully convolutional fashion[1,7,17]. Inspired by "Atrous" scheme of [1], we modify the ResNet101 model (released in [25]) as our baseline model. We replace the 1 000-way ImageNet classifier in the last layer with a 151-way Softmax classifier (150 semantic classes and one extra background class). The loss is the sum of cross-entropy for each spatial position in the CNN output map. The FEN (feature extraction network) is fine-tuned on the SceneParse150 dataset. Fig.3 show the detailed architecture of the proposed OENet. OENet is composed of three parts. The whole image is first fed into several convolutional layers to generate feature maps, and this network is a ResNet101-like structure and modified by multi-level multi-Scale image representations (see Subsection 3.1). Then these feature maps are sent to objectness proposal network (branch 2) for locating and recognizing objects. Each image can contain multiple objects, and each object includes the prediction of object position and category (see Subsection 3.2). Next, the features along with the object proposals are passed into a sub-network (box-level instance enhancement network (branch 1) or mask-level instance enhancement network (branch 3)) to generate the confidences enhanced feature maps (see Subsection 3.3). Finally, fully-connected CRF and black-hole filling strategies are employed to optimize the parsing results (see Subsection 3.3.4 and Subsection 3.4).

3.1 Feature Extraction Network

CNN has shown a remarkable ability to implicitly represent the scale of an object by training on the images with diverse scales. Besides, explicitly considering multi-scale in design can improve the recognition performance for both large and small objects. We expand our baseline model to a multi-scale version through integrating a multi-level multi-scale strategy. Fig.4 illustrates the shared convolutional layers of our multi-scale feature extraction network (FEN).

We investigate two approaches to manage scale variability in scene parsing. Firstly, a generic multi-scale processing method[34] is used to process input images. We extract convolutional feature maps from three different scales. More specifically, the original image is resized by a fixed factor *f*, and then is propagated by parallel CNN branches. All three branches share the same structure and parameters. To produce the finer feature map, we bilinearly interpolate the feature maps from the parallel CNN branches to the original image resolution and fuse them by taking at each position the maximum response across different scales. We are doing this during both training and testing. Secondly, a spatial pyramid pooling[31] method applied on "Atrous" convolutional layer is incorporated into our baseline model. We use multiple parallel atrous convolutional layers with different dilation rates to produce different

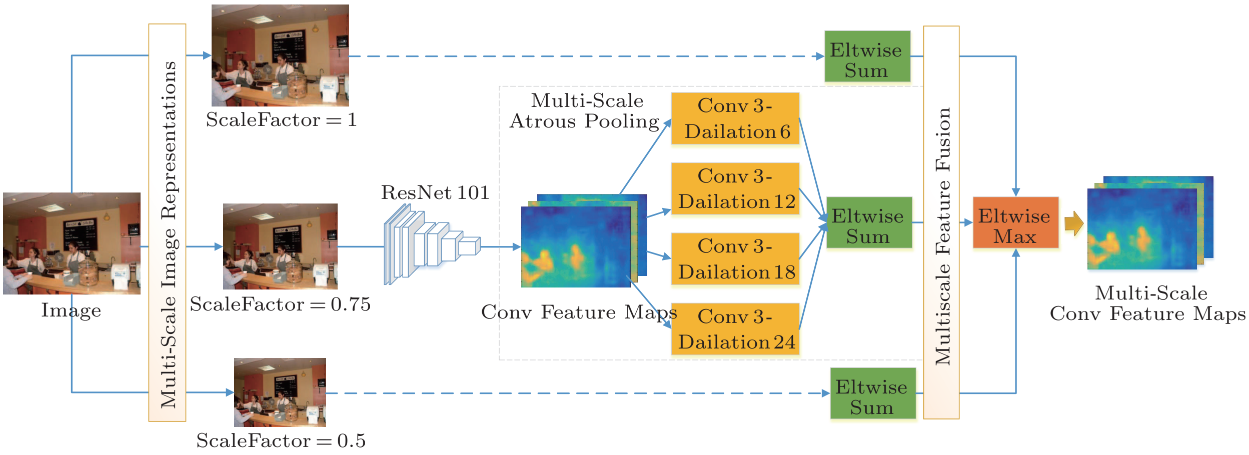

Fig.4. Multi-level multi-scale image representations.

single-scale feature maps. All the single-scale feature maps are further fused to generate the final multi-scale feature map. We call this two-level multi-scale CNN as feature extraction network (FEN). The output of FEN can be used to generate region proposals (Subsection 3.2) and objectness instances (Subsection 3.3) simultaneously. The final multi-scale feature can be formulated as:

$$\mathcal{F}_{\text{Multi}} = \max_{m\in[1,\ldots,M]} \sum_{n=1}^{N} \mathcal{F}_{(m,n)},$$

where $\mathcal{F}_{\text{Multi}}$ denotes the final multi-scale feature, and $\mathcal{F}_{(m,n)}$ denotes the single-scale feature which is produced from branch m with resolution R_m and branch n with dilation rate k_n. We denote R_m as the image resolution in multi-scale resolution branch $m = 1, 2, \ldots, M$. Each R_m has a fixed scale factor $f \in \{0.5, 0.75, 1\}$, and the resolution of branch m can be represented as $R_m = f \times R_{\text{in}}(width, height, 3)$, where $R_{\text{in}}(width, height, 3)$ is the original input resolution. Meanwhile, we denote $k_n \in \{6, 12, 18, 24\}$ as the dilation rate in multi-scale atrous pooling branch $n = 1, 2, \ldots, N$. The total number of branches m and n is set to 3 and 4 in this paper respectively.

3.2 Objectness Proposal Network

Recognition and boundary errors are both the key problems of scene parsing as described in [35]. Recognition errors occur when object categories are recognized incorrectly or missing. As shown in Fig.5, the tree and the bus in row 1 and the tricycle in row 3 are missing, and the chair and the desk in row 2 are recognized as wrong categories. Our objectness enhancement is designed for recalling these missing objects. On the other hand, fully-connected CRF[36] is used to handle the boundary errors that occur when semantic labels are incorrect at the edges. To ensure integrity, we briefly introduce the fully-connected CRF in Subsection 3.3.

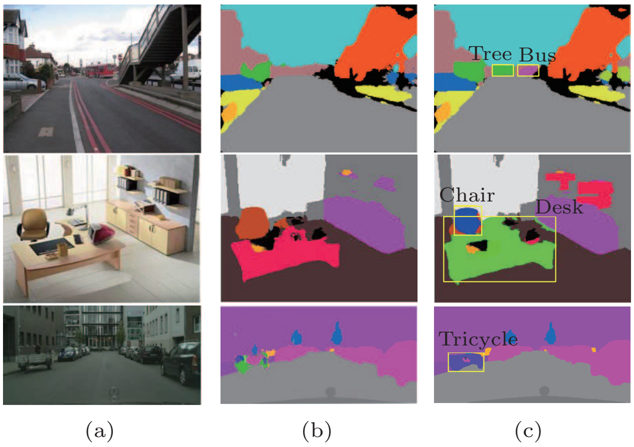

Fig.5. With and without objectness enhancement for scene parsing. (a) Original image. (b) Segmentation without objectness enhancement. (c) Segmentation with objectness enhancement. This figure shows how we improve the parsing results about the bus, chair, desk, and tricycle.

Inspired by [12-14], objectness proposal network (OPN) is used here to help separating objects from overlapping ones and complex stuffs, which makes the model focused on instance-level objects. In this paper, we use "objectness" as our measure indicator, which not only indicates the region of proposal objects, but also specifies the category of proposal objects. OPN is behind FEN, and uses its multi-scale convolutional feature maps as input for objectness proposal and semantic understanding simultaneously. We use different regions of an input image to represent the receptive fields of different prediction objects. The classification and the bounding box regression are performed to estimate locations and category scores of these objectness regions. For training OPN, the ground-truth used here is the circumscribed rectangle of object segmentation ground-truth, and thus no extra information is used. In this paper, we collect 115 discrete object classes (i.e., car, person, table) from the SceneParse150 dataset[2] for training the objectness proposal network. Together with additional 35 stuff classes, 150 classes are used to evaluate models on the scene parsing task. This definition is the same with the standard of ScenePares150 benchmark[2].

The network structure and the loss function of this stage follow the work of region proposal networks (RPN)[13] and Fast RCNN detection network[12], which we briefly describe as follows for completeness. Both RPN and detection network predict the locations of bounding boxes and object scores in a fully-convolutional form. The difference is that the bounding boxes of RPN are class-agnostic, while they are associated with categories in detection. The branch (2) of Fig.3 shows the framework of OPN. On top of the shared convolutional layer, a 3×3 convolutional layer is employed to reduce dimensions, followed by two sibling 1×1 convolutional layers for regressing the locations of bounding boxes and classification. We utilize a multi-task loss function to train OPN. For each anchor, its loss function is defined as:

$$\begin{aligned} & L(R(k, k^*, t, t^*)) \\ = & \frac{1}{N_{\rm cls}} L_{\rm cls}(k, k^*) + \lambda \frac{1}{N_{\rm reg}} k^* L_{\rm reg}(t, t^*), \end{aligned} \quad (1)$$

where the ground-truth class label k^* is 1 if the anchor is positive, and 0 if the anchor is negative. Moreover, $k \in (k_0, k_1, \ldots, k_K)$ is a discrete probability distribution over $K+1$ categories (the additional 1 is the extra background class, and denoted as k_0). For simplicity, we implement the classification as a two-class softmax layer. Alternatively, one may use logistic regression to produce k scores. Thus, $L_{\rm cls}(k, k^*) = -\log p_{k^*}$ is the standard cross-entropy loss for the classification over two classes (object vs non-object). The second term, loss $k^* L_{\rm reg}(t, t^*)$ is defined over a tuple of true bounding box regression for class k^*, and it is activated only for positive anchors ($k^* = 1$) and is shielded for others ($k^* = 0$). In addition, $t^* = (t^*_x, t^*_y, t^*_w, t^*_h)$ denotes the ground-truth bounding box, and a predicted tuple $t = (t_x, t_y, t_w, t_h)$ again for class k^*. Finally, $L_{\rm reg}(t, t^*) = \sum_{i \in x,y,w,h} R(t_i - t^*_i)$, where

$$R(*) = {\rm smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise}, \end{cases}$$

is a robust smoothed L_1 loss function defined in [12] that is less sensitive to outliers than L_2 loss used in R-CNN[27]. The two terms are normalized by $N_{\rm cls}$ and $N_{\rm reg}$ and weighted by a balancing parameter λ. In our implementation, the cls term $L_{\rm cls}(k, k^*)$ in (1) is normalized by the mini-batch size (i.e., $N_{\rm cls} = 64$ in VGG16) and the reg term $L_{\rm reg}(t, t^*)$ is normalized by the number of anchor locations (i.e., $N_{\rm reg} = 2\,400$). By default, we set $\lambda = 10$, because the reg term is more important than the cls term in regional proposals. Although other values may be more suitable for training a better OPN, λ is not a decisive parameter. Finally, R is the output of OPN, represented as a list of boxes $R_i = \{x_i, y_i, w_i, h_i, p_i, c_i\}$, where i is the index of R_i. R_i is centered at (x_i, y_i) with width w_i and height h_i, and p_i is the probability of category c_i. For simplicity, R_i can be denoted as $R_i = \{t_i, p_i, c_i\}$, where $t_i = \{x_i, y_i, w_i, h_i\}$.

3.3 Objectness Enhancement Network

In our objectness enhancement network (OEN), the network takes multi-scale features as the input, and outputs box-level or mask-level instance-aware semantic segmentation results. The cascade OEN consists of three stages: box-level instance enhancement, mask-level instance enhancement, and fully-connected CRF. We use the category-based region proposal which is the output of OPN as the auxiliary input, and combine it with the multi-scale features to calculate the final results. As shown in Fig.3 and Fig.6, we produce two different parsing results. 1) The box enhanced parsing is derived from the box-level instances and multi-scale convolutional feature maps. 2) The mask enhanced parsing is derived from the mask-level instances and multi-scale convolutional feature maps. The mask-level instances can be calculated by the box-level instances and multi-scale convolutional feature maps. In this subsection, we sequentially introduce these techniques: box-level instances, mask-level instances, instance enhancement, and fully-connected CRF.

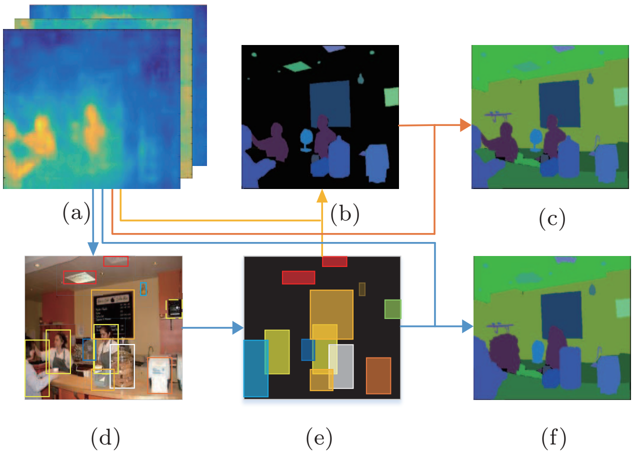

Fig.6. Flowchart of objectness enhancement. (a) Convolutional feature maps. (b) Mask-level instances. (c) Mask enhanced parsing. (d) Region proposals. (e) Box-level instances. (f) Box enhanced parsing.

3.3.1 *Box-Level Instances*

Box-level instances are rough collections of objectness. As mentioned before, the region proposal $R_i = \{t_i, p_i, c_i\}$ is formed by a location and category. Because the object is encircled by the region proposal, we can roughly consider the rectangular area $t_i = \{x_i, y_i, w_i, h_i\}$ as a box-level instance, and it corresponds to category c_i. That is to say, all the pixels in region t_i are regarded as a part of the box-level instance. We denote the box-level instance $B_i(W, H, c_i)$ as:

$$B_i(W,H,c_i) = \begin{cases} 1, \text{ if } p \in R_i(t_i), \\ 0, \text{ otherwise}, \end{cases}$$

where p is a pixel of $B_i(W, H, c_i)$, and W and H are the width and the height of the box-level instance respectively.

3.3.2 *Mask-Level Instances*

Objects are usually irregular in nature, and describing an object by a rectangle will be mixed with a large number of background pixels. Simply using the box-level instances can improve the identification ability of objects, but will damage the identification ability of stuffs. As shown in Fig.3, we combine the feature maps and box-level instances to produce mask-level instances. For each box-level instance $B_i(W, H, c_i)$, we can get one mask-level instance $M_i(W, H, c_i)$. This process can be formulated as:

$$\begin{aligned} & M_i(W,H,c_i) \\ = & \begin{cases} 1, \text{ if } p \in F(W,H,c_i) \times R_i(t_i,c_i) > t, \\ 0, \text{ otherwise}, \end{cases} \end{aligned} \quad (2)$$

where $F(W, H, c_i)$ is the feature map corresponding to category c_i. Width W and height H are the same with those of the input image. Threshold t controls the size of the mask, and the upper boundary is the size of box-level instance. It can be drawn from (2) that the mask is only related to the specific feature map, which has the same category with region proposals. Besides, one category may have many instance objects, and thus we can group the mask-level instances of these objects into a feature map for simplifying the calculation. Further, we can get the entire feature maps $M(W, H, C)$ through iterating all R_i, where $C = 1, ..., c_i$ is the category space.

3.3.3 *Instance Enhancement*

After obtaining the box-level or mask-level instances, we can generate the objectness enhancement feature from the multi-scale convolutional feature map. In this paper, we combine instance regions and feature maps with tied weighs strategy. The pixels which belong to the instance regions have been enhanced by weight w and probability p_i. The mask-level enhancement feature $F_{\rm ME}$ can be defined as:

$$\begin{aligned} & F_{\rm ME}(W,H,C) \\ = & F(W,H,c_j) \times M_i(W,H,c_i) \times w \times p_i \times c^*, \end{aligned} \quad (3)$$

where $c^* = 1$ if $c_i = c_j$, and zero otherwise. This means that objectness enhancement being activated only happens when the feature map and the object instance have the same label. In (3), replacing $M_i(W, H, c_i)$ with $B_i(W, H, c_i)$ can get the box-level enhanced feature $F_{\rm BE}$.

3.3.4 *Fully-Connected CRF*

Due to the multiple max-pooling layers, the increased invariance and the large receptive fields can yield quite smooth responses and homogeneous classification results in scene parsing. To overcome these limitations, we integrate the fully-connected CRF model[1,36] into our OENet to refine the feature map as a postprocessing stage. Our model employs the energy function:

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j),$$

where x is the label of pixels in an image. The unary potential $\theta_i(x_i) = -\log P(x_i)$, and $P(x_i)$ is the inference probability at pixel i as computed by CNN. The second pairwise potential allows for the efficient inference between a pair of connecting images by a fully-connected graph. We denote the pairwise potential as follows:

$$\begin{aligned}&\theta_{ij}(x_i, x_j)\\ =\ &\mu(x_i, x_j)(\lambda_1 \exp(-\frac{||p_i - p_j||^2}{2\delta_\alpha^2}) +\\ &\lambda_2 \exp(-\frac{||p_i - p_j||^2}{2\delta_\beta^2} - \frac{||c_i - c_j||^2}{2\delta_\gamma^2})).\end{aligned}$$

We denote i and j as the position of a pixel in the convolutional feature maps and its original input image. If $x_i \neq x_j$, $\mu(x_i, x_j) = 1$, and zero otherwise. That is to say, only nodes with different labels are penalized. Two Gaussian kernels are used in different feature spaces. The former exp term in the above equation indicates only pixel position is valid, and it only considers spatial proximity when enforcing smoothness; the latter exp term indicates a "bilateral" kernel depends on both RGB color (denoted as c) and pixel position (denoted as p), and it forces pixels with a similar color and position to have similar labels. The hyper parameters $\delta_\alpha, \delta_\beta$ and δ_γ control the scale of Gaussian kernels, and the weight parameters λ_1 and λ_2 are used to balance the two features.

In this paper, all the experiments are conducted with fully-connected CRF.

3.4 Filling Black-Hole

The "extra background" category, as a specific category, is often attached to the category space in detection and segmentation tasks. It can significantly improve the performance. This strategy can remove the ambiguous samples from the positive gallery, which makes the classifier more robust. However, in scene parsing task, some pixels will be assigned to the "extra background" category in the inference stage. This is an obvious classification error. We call this problem as "black-hole". As shown in (4), we develop a very intuitive way to overcome this issue. The output label can be denoted as:

$$O_{L(i,j)} = \mathop{\arg\min}_{c\in[2,\dots,N]} O_{F(i,j,c)}, \tag{4}$$

where $O_{F(i,j,c)}$ is the feature computed by CNN, (i, j) is the position of the pixel, and c is the number of the feature channel which is equal to the number of categories with an "extra background" category. We denote $c = 1$ as the "extra background" category, and the rest are the other categories. In general, we assign the channel ID which has the maximum probability in all channels as the label. In this paper, we rule out the channel $c = 1$ (extra background), then calculate the maximum probability of the rest feature maps, and set the ID as the prediction label. In other words, we repair the "black-hole" regions by specifying the label which has the second largest probability if the index of predicting category equals 1. This strategy significantly increases the parsing accuracy.

4 Experiments

4.1 SceneParse150

4.1.1 Dataset

We evaluate the proposed OENet framework on the SceneParse150 scene parsing benchmark dataset. The original dataset contains 20 210 (train), 2 000 (val) pixel-level labeled images for training and validation, respectively. The performance is measured in terms of pixel intersection-over-union averaged across the 150 classes (mIoU, mean intersection-over-union) and the proportion of correctly classified pixel (pixel accuracy). For training the region proposal network, we extract the bounding box ground-truth of an object from the circumscribed rectangle of segmentation ground-truth. Among the 150 classes, there are 35 stuff classes (i.e., wall, sky, road) and 115 object classes (i.e., car, person, table). The bounding box ground-truth of an object is produced only from the 115 object classes.

4.1.2 Implementation Details

For training OENet, we employ the ResNet101 network pre-trained on MSCOCO[9] dataset, and a 5-step training process for optimization as shown in Algorithm 1. For training FEN and OEN, we employ "ploy" learning rate policy (the learning rate is multiplied by $(1 - iter/max\_iter)^{power}, power = 0.9$), and use a mini-batch of 10 images and initial learning rate of 0.002 5. We utilize momentum of 0.9 and weight decay of 0.000 5. After FEN has been fine-tuned on the training set, we cross-validate the CRF parameters according to [1]. We employ 10 mean field iterations. We use default values of $\lambda_1 = 3$ and $\delta_\alpha = 3$ and search for the best values of λ_2, δ_β, δ_γ by cross-validation on 200 images from validation. We employ a coarse-to-fine search strategy. The initial search range of the parameters is $\lambda_2 \in [3 : 6], \delta_\beta \in [3 : 6]$, and $\delta_\gamma \in [30 : 10 : 100]$, and then we refine the search step sizes around the first round's best values. We employ "step" learning rate policy to train OPN, and 40 000 and 80 000 iterations are enforced for RPN and detection network respectively. The initial learning rate is 0.001. All the networks use the momentum of 0.9 and the weight decay of 0.005. During benchmarking on the multi-scale network, we downsample the images into a low resolution. We set the long side at 500 pixels, and for augmentation, the input size in the training and test protocol is 513 pixels and 321 pixels, respectively. All experiments are performed in the open source framework CAFFE[37] with NVIDIA Titan X GPU.

Algorithm 1. Training Process OENet

Step 1: Pre-train a deep CNN model on the MSCOCO[9] dataset, as described in [9].

Step 2: Train the feature extraction network (FEN) with a cross-entropy classifier for each spatial position on the target dataset. This network is initialized with the pre-trained model in step 1. It is used for initializing other networks, and it is also used as our baseline model.

Step 3: Cross-validate the CRF parameters according to [1].

Step 4: Train objectness proposal network (OPN), which is initialized with an extraordinary model pre-trained in step 2. In this step, we first train the RPN subnet, and then use the proposals generated by RPN to train the detection subnet. The shared convolution layers are always fixed.

Step 5: Output the unified OENet trained in step 2 and step 4, the CRF modular, the region enhancement modular and the black-hole filling modular are also integrated into the entire network.

4.1.3 Ablation Studies

We evaluate the effectiveness of the four important components of OENet. The performance over all 150 categories from five variants of OENet is reported in Table 1. With the same training protocol, multi-level multi-scale brings 4.1% and 1.5% improvements on mIoU and pixel accuracy, respectively. As analyzed in Section 3, objectness enhancement is beneficial to recall the missing objects, which results in performance promotion in mIoU by 1.5%. Employing the black-hole filling strategy for post-processing, our final output substantially outperforms the model without this strategy by 1.9% and 2.2% on mIoU and pixel accuracy, respectively.

In order to further validate the effectiveness of our proposed method, we study the IoU details of each category. Detailed results are listed in Appendix A. First, we find that the performance improvement has no bias. That is to say, both the objects and stuffs have been improved, although our approach focuses on processing objectness regions. Second, the performance of 109 categories is improved over multi-scale model in the entire dataset, accounting for 72.7%. These results are also class-agnostic. Third, after objectness enhancement, there are 70.7% categories overtaking the multi-scale model. The failure cases include 15 stuffs and 29 objects, but most of the objects are usually considered as the background, i.e., a tree, fence, column, bathtub, and so on. These results verify the effectiveness of the objectness region enhancement. Finally, with black-hole filling technology, 60% categories again achieve better results. In general, 88% categories outperform the baseline model with our OENet model.

Table 1. Ablation Studies on the Validation Set of SceneParse150

MMS	Box	Mask	BH	mIoU	Pixel Accuracy (%)
				30.9	74.0
√				35.0	75.5
√	√			35.6	75.1
√	√	√		36.5	75.7
√	√	√	√	38.4	77.9

Note: MMS: muti-level multi-scale, Box: box-level region enhancement, Mask: mask-level region enhancement, and BH: black-hole filling.

4.1.4 Region Proposal Evaluations

We also concern about how the region proposal policy affects the performance in the scene parsing task. In this subsection, we make a comparison with some systems that are designed for object detection. We train OPN under the FasterRCNN[13] and RFCN[14] framework to produce box-level instances for our OENet. FasterRCNN-based and RFCN-based OPN use the same convolutional framework, but use different detection modules. Besides, the bounding box ground-truth is used as the upper boundary for evaluating. As shown in Table 2, same with the detection results, the parsing results on the FasterRCNN framework are slightly better than those on the RFCN framework. It is encouraging that the parsing results running on bounding box ground-truth achieve ideal results by mIoU of 47.8 and pixel accuracy of 80.3% on validation set. This upper boundary means that our proposed method has room for improvement.

Table 2. Effect of Different Detection Methods

Method	Detection	mIoU	Pixel
	Accuracy (%)		Accuracy (%)
GT	100.0	47.8	80.3
FasterRCNN[13]	84.4	38.4	77.9
RFCN[14]	82.3	38.0	77.7

4.1.5 Comparison with State-of-the-Arts Methods

As for the baseline of scene parsing on the SceneParse150 benchmark, several state-of-the-art methods[1-2,7,16,38] and a baseline model which is modified from the ResNet101 model are used for comparison. FCN[7] upsamples the activations of multiple layers in CNN for pixel-wise segmentation. SegNet[16] is a encoder and decoder architecture used for image segmentation. DilatedNet[38] drops pool4 and pool5 from fully convolutional VGG16 network, and replaces the following convolutions with dilated convolutions. Cascade-SegNet and Cascade-DilatedNet[2] construct a cascade multiple stream model to generate stuff, object and part maps from shared feature activation, and then merge all the maps to produce full scene parsing. DeepLabv2[1] is a multi-scale ResNet101 model with atrous spatial pyramid pooling and fully-connected CRF. Our baseline model is based on the DeepLab model, but without multi-scale and ASPP scheme.

We report the evaluation results in Table 3. With all the components, our final model yields 38.4 mIoU and 77.9% pixel accuracy, which significantly outperform the baseline by 7.5% and 3.9% on the validation dataset respectively. These results are also better than those of the other comparison models.

Table 3. Comparsion Among OENet with State-of-the-Art Methods on the Validation Set of SceneParse150

Method	mIoU	Pixel Accuracy (%)
SegNet[16]	21.6	71.0
Cascade-SegNet[2]	27.5	71.8
FCN8s[7]	29.4	71.3
DilatedNet[38]	32.3	73.6
DeepLabv2[1]	34.3	75.3
Cascade-DilatedNet[2]	34.9	74.5
Baseline	30.9	74.0
OENet	**38.4**	**77.9**

4.1.6 Qualitative Results

We visualize the parsing results of our baseline model with mask-level objectness region enhancement, black-hole filling strategy in Fig.7. The baseline model loses some objects, and our model is able to recall some missing objects through employing objectness region enhancement. Benefiting from the local contextual information, OPN gathers the related pixels and combines them into objectness, especially to discover some objects which are small (e.g., the light in row 2 and the tea table in row 6) or overwhelmed by a large area of the background (e.g., the red bus and white van in row 1, the chair in row 3). The advantage of black-hole filling is also very obuvious. It recalls the pixels which are assigned to the extra background class. These above errors look like a black hole. Our algorithm tries to find an appropriate category to populate it. Rediscovering the ground in row 1, the car in row 4, and the sash door in row 5 is favorable evidence.

4.1.7 Failure Cases

As mentioned above, objectness enhancement and black-hole filling bring some good properties; however they are not always effective. The major problems include three aspects. 1) Species in nature are complex and diverse, and thus for those objects that are not existing in the training set, both the classifier and detector are unable to identify them. The regions of these objects may be identified as the surrounding background. As shown in row 1 of Fig.8, the fishes are missing in the water after black-hole filling. 2) Because the objects and the background are very similar, the classifier and the detector may be deceived by the visual sense. On one hand, the segmentation network may output incorrect classification results. On the other hand, the detection network may not be able to find the target. As shown in row 2 of Fig.8, the bus has not been fully recognized by the baseline segmentation network. Besides, it also has not been detected by our objectness proposal network. The results of three models are consistent. 3) For the complex scenes, especially the scenes with crossing and overlapping objects, the classifier and the detector will feel rather confused. Such as the example of row 3 in Fig.8, the view is obscured by the iron gate. This phenomenon makes all the results quite messy.

4.2 Cityscapes

4.2.1 Dataset

Cityscape[39] is a recent large-scale high-resolution scene understanding dataset which contains high quality pixel-level annotations of 5 000 images collected from 50 cities in different street scenes. It defines 19 categories containing both objects and stuffs. In all, the training, the validation, and the test set contain 2 975, 500, 1 525 images, respectively.

4.2.2 Implementation Details

Same with the setting of the SceneParse150 dataset, we employ the ResNet101 network for extracting fea-

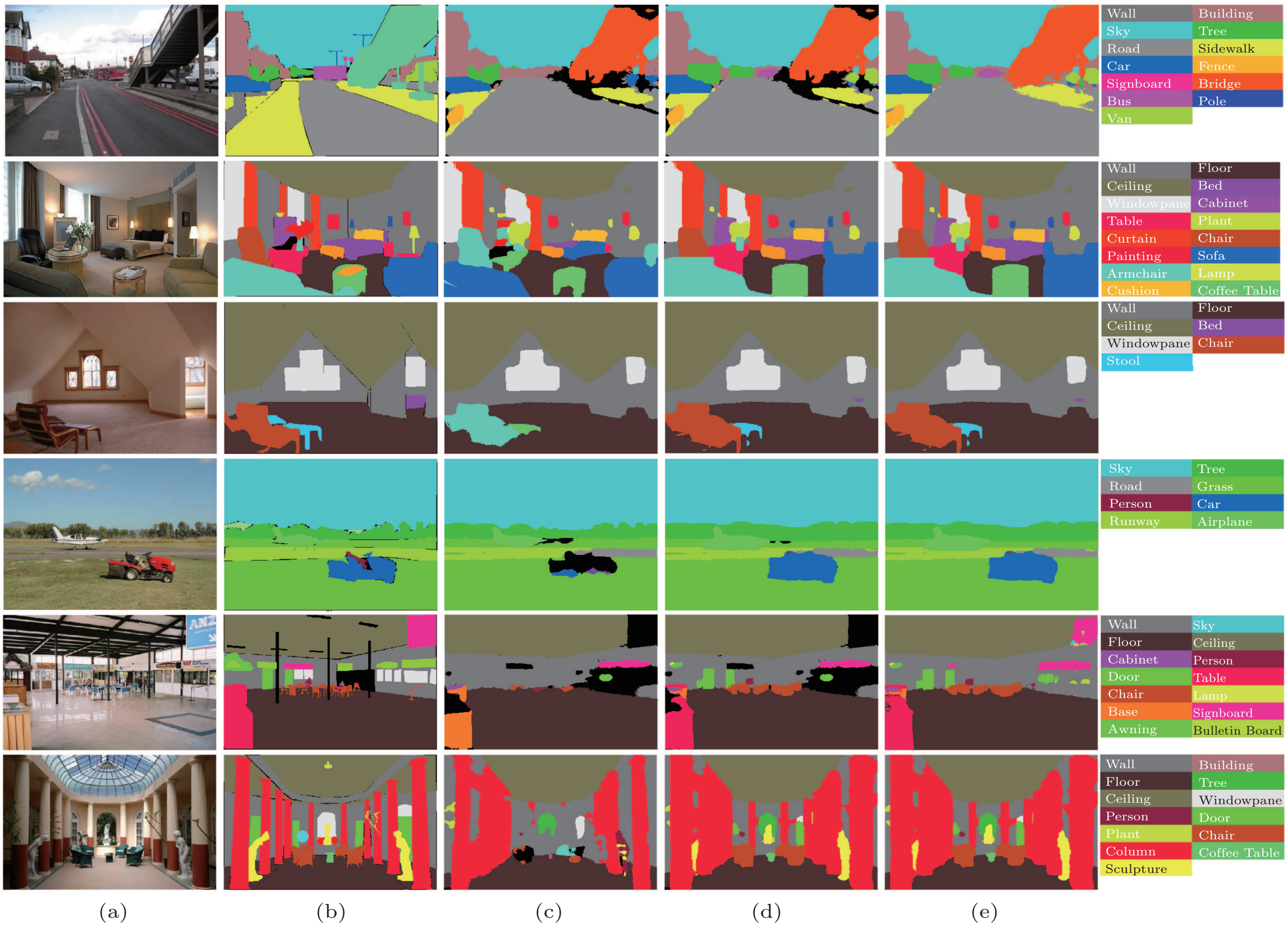

Fig. 7. Examples of scene parsing on the SceneParse150 dataset. (a) Input image. (b) Ground-truth. (c) Baseline. (d) Segmentation with multi-scale representations and objectness enhancement. (e) Segmentation with multi-scale representations, objectness enhancement, and black-hole filling.

tures. We do not exploit multi-scale image representation due to the limited GPU memory, but retain multi-scale atrous pooling. The high-resolution (2 048×1 024) is a challenging problem for training deep network with the limited GPU memory. In order to solve this problem, we crop the original image into 705×705 overlapped patches, and train on these patches without downsampling. Each image is split into eight patches. For data augmentation, the input sizes of the training and test network are 545 pixels and 705 pixels, respectively. Other hyper-parameters are the same with the setting of the SceneParsing150 dataset.

4.2.3 *Results on Validation Set*

We explore the validation set in Table 4. With the same training protocol, results on high-resolution significantly bring 2% and 3.1% improvements before and after objectness region enhancement, respectively. Employing our OENet method brings 0.7% and 1.8% improvement without and with high-resolution training respectively. We conclude that high-resolution training helps the objectness proposal network to find smaller objects. Therefore, OENet shows more obvious advantages in the parsing task with high-resolution images.

4.2.4 *Results on Test Set*

We upload our baseline model and high-resolution OENet model to the evaluation server (Table 5), obtaining performance of 69.8% and 71.3%, respectively. Although our OENet does not have the best performance, it is still competitive. More importantly, our core algorithms including objectness region enhancement and black-hole filling strategies have been shown to be effective in the scene parsing task. Note that our model is only trained on the training set, and we do not use the coarse annotation. The results are reported on a single

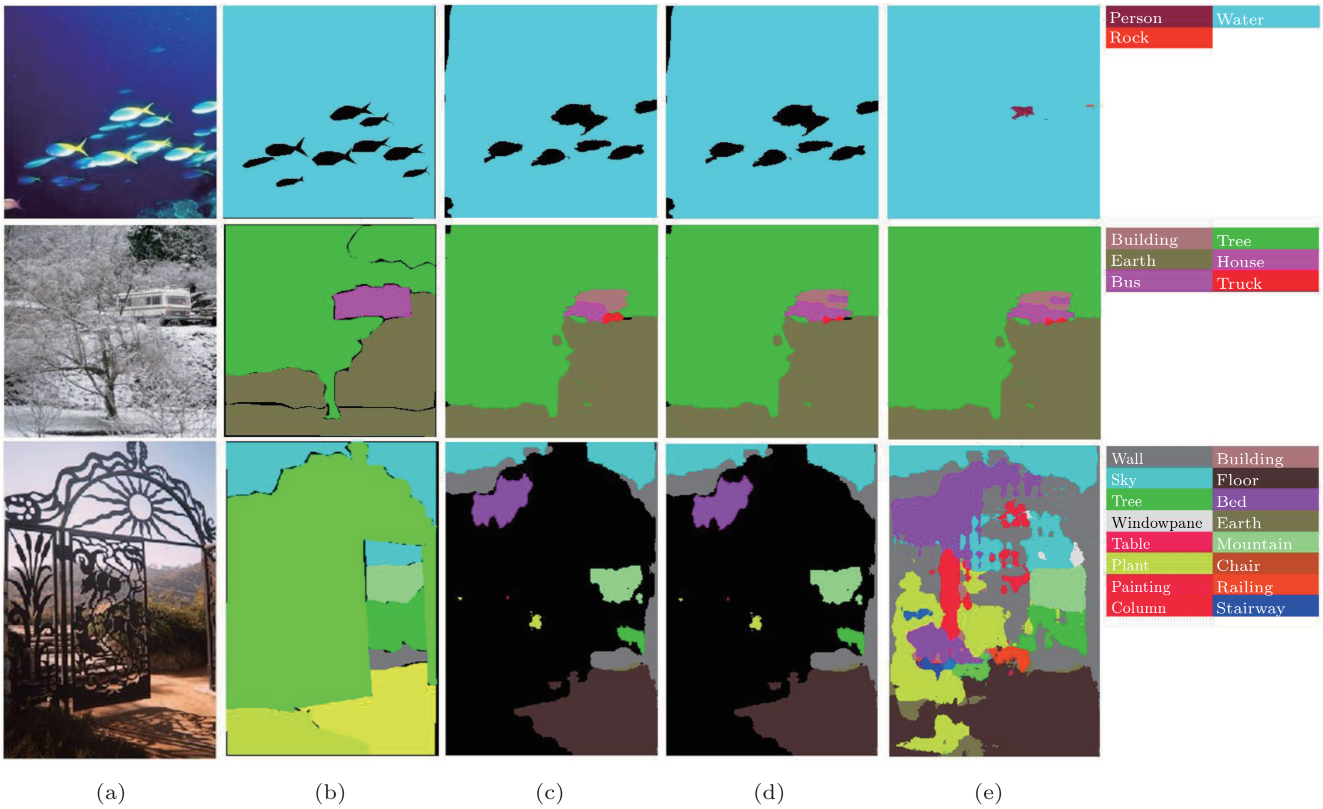

Fig.8. Three failure cases on the SceneParsing150 dataset. (a) Input image. (b) Ground-truth. (c) Baseline. (d) Segmentation with multi-scale representations and objectness enhancement. (e) Segmentation with multi-scale representations, objectness enhancement, and black-hole filling.

model, and we do not use the multi-scale training and test strategies.

4.2.5 *Qualitative Results*

We visualize the results in Fig.9. Benefiting from the objectness enhancement, the integrity of an object is well maintained (e.g., the person in row 2, the tail of the car in row 3, and the truck in row 4 and row 6). We find high-resolution images also help to improve the performance of objectness enhancement. As shown in row 1 and row 5 of Fig.9, our method finds some smaller objects (e.g., the car in row 1 and the motorcycle in row 5).

4.2.6 *Failure Cases*

As mentioned previously, objectness enhancement can preserve the integrity of an object. However, objectness enhancement may also destroy some crossing objects. As shown in Fig.10, comparing Baseline-HR with OENet, the poles (before the car in row 1 and row 2, before the rider in row 3) and the vegetation (among the right cars in row 4) are lost in the background objects.

Table 4. Results on Cityscapes Validation Set

	Method	mIoU
VGG16	DeepLabv2-VGG16[1]	62.9
	FCN[7]	63.4
	Pixel-level encoding	64.3
	DPN[40]	66.8
	DilatedNet[38]	67.1
	Adelaide[8]	68.6
ResNet101	DeepLabv2-Resnet101[1]	71.4
	Baseline	69.3
	Baseline-HighResolution	71.3
	OENet	70.0
	OENet-HighResolution	**73.1**

Note: HighResolution: train and test on 705×705 high-resolution patches.

5 Conclusions

In this paper, we proposed and "OENet" for scence parsing, which is trained on image classification network. In order to recall the missing objects, an objectness proposal network based objectness enhancement was proposed to produce box-level instances and mask-

Table 5. Performance Comparison of OENet with the State-of-the-Art Methods on Cityscapes Test Set

	FCN8s[7]	DPN[40]	DilatedNet[38]	DeepLabv2[1]	Adelaide[8]	Baseline	OENet
Road	97.4	97.5	97.6	97.9	**98.0**	97.4	97.5
Sidewalk	78.4	78.5	79.2	81.3	**82.6**	78.3	79.4
Building	89.2	89.5	89.9	90.3	**90.6**	90.0	90.5
Wall	34.9	40.4	37.3	48.8	44.0	45.8	**48.5**
Fence	44.2	45.9	47.6	47.4	**50.7**	46.3	49.0
Pole	47.4	51.1	**53.2**	49.6	51.1	40.6	43.5
Traffic Light	60.1	56.8	58.6	57.9	**65.0**	53.4	55.7
Traffic Sign	65.0	65.3	65.2	67.3	**71.7**	65.6	67.3
Vegetation	91.4	91.5	91.8	91.9	**92.0**	91.3	91.7
Terrain	69.3	69.4	69.4	69.4	**72.0**	67.6	69.2
Sky	93.9	94.5	93.7	94.2	94.1	94.5	**94.8**
Person	77.1	77.5	78.9	79.8	**81.5**	79.8	80.8
Rider	51.4	54.2	55.0	59.8	61.1	59.2	**61.2**
Car	92.6	92.5	93.3	93.7	**94.3**	93.9	94.2
Truck	35.3	44.5	45.5	56.5	61.1	62.8	**64.6**
Bus	48.6	53.4	53.4	57.5	65.1	69.3	**70.8**
Train	46.5	49.9	47.7	57.5	53.8	62.5	**64.4**
Motorcycle	51.6	52.1	52.2	57.7	**61.6**	59.1	**61.1**
Bicycle	66.8	64.8	66.0	68.8	**70.6**	68.5	70.0
mIoU	65.3	66.8	67.1	70.4	**71.6**	69.8	71.3

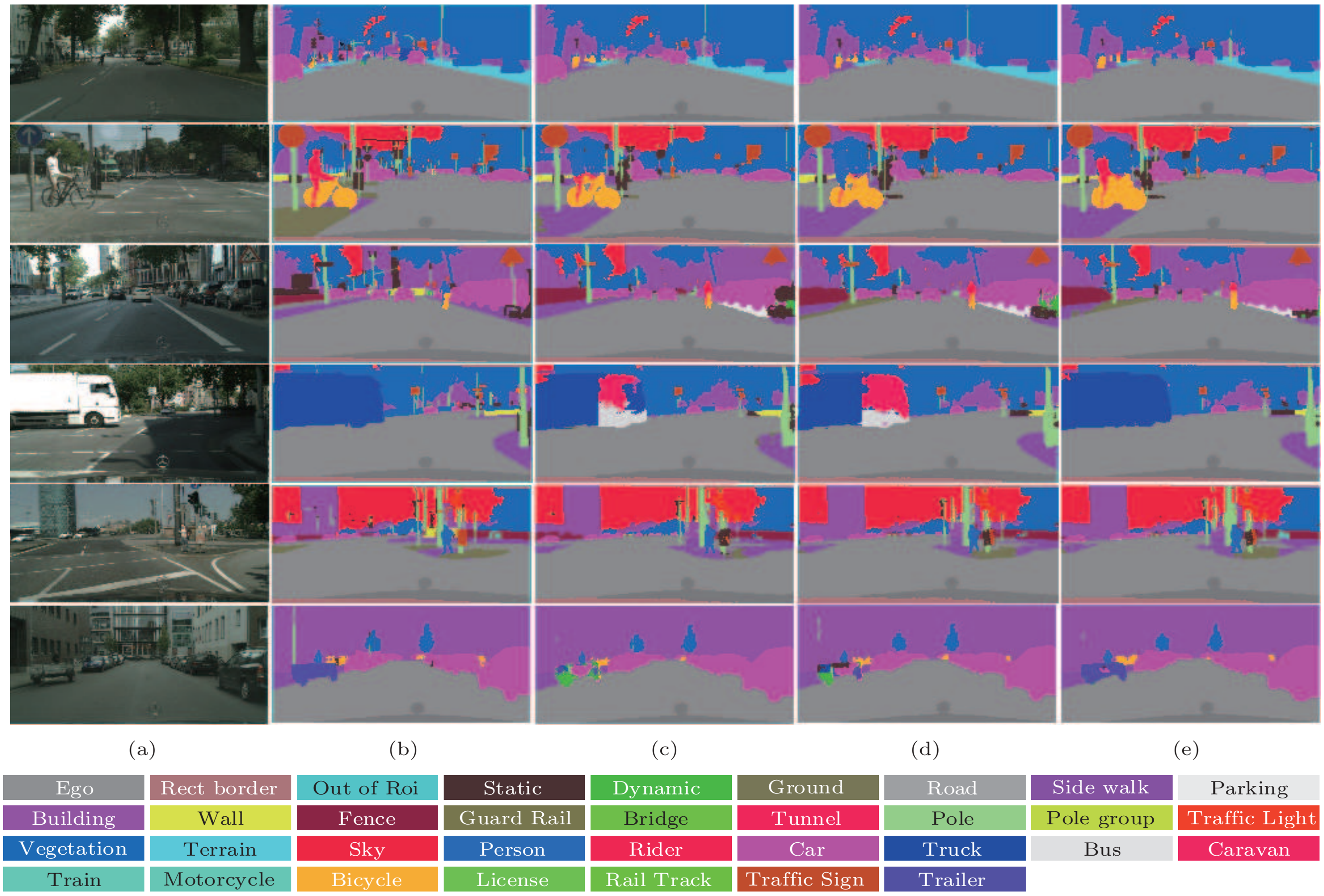

Fig.9. Examples of scene parsing on the Cityscapes dataset. (a) Input image. (b) Ground-truth. (c) Baseline. (d) Segmentation with high-resolution representations. (e) Segmentation with OENet with objectness enhancement, black-hole filling strategies, and high-resolution representations. Roi: Region of interest.

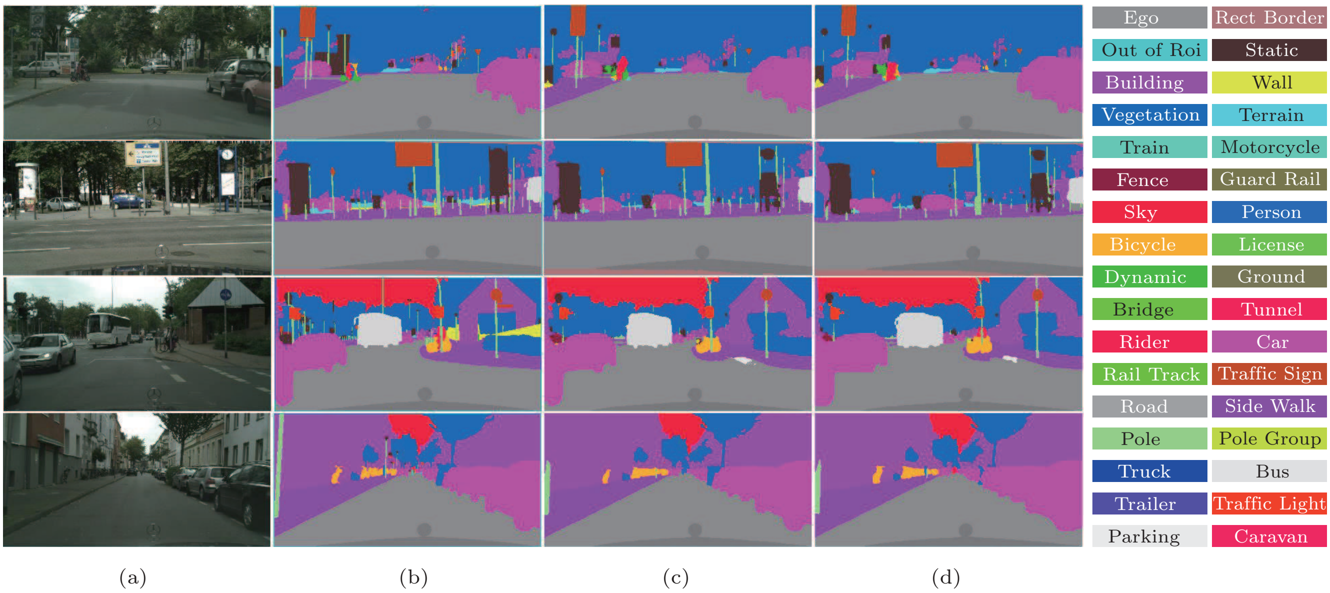

Fig.10. Four failure cases on Cityscapes Dataset. (a) Input image. (b) Ground-truth. (c) Segmentation with high-resolution representations. (d) Segmentation with OENet with objectness enhancement, black-hole filling strategies, and high-resolution representations.

level instances. With objectness enhancement strategy, some missing objects are recalled. Both box-level instances and mask-level instances can be considered as objectness. The only difference is that the mask-level instance can be regarded as a fine-grained box-level instance. Therefore, we could use the box-level instances and the convolutional feature maps to synthesize the mask-level instances. To produce semantically accurate predictions and detailed parsing results along object boundaries, we also combined ideas from deep convolutional neural networks and full-connected CRF. Finally, the black-hole filling strategy effectively processes those pixels misallocated to the superfluous extra background class. Our experimental results showed that the OENet method significantly outperforms the state-of-the-art methods on the challenging datasets SceneParse150 and Cityscapes. It should be noted that our core algorithms, objectness region enhancement, and black-hole filling techniques are not limited in OENet, and they can be embedded into other parsing networks as separate modules to improve the parsing capacity in the objectness area. In the future, we will explore how to further improve the performance of the objectness region proposal, thereby improving the segmentation results.

References

[1] 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. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017, doi: 10.1109/TPAMI.2017.2699184.

[2] Zhou B L, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A. Semantic understanding of scenes through the ADE20K dataset. *arXiv: 1608.05442*, 2016. https://arxiv.org/abs/1608.05442, June 2017.

[3] Fu Z J, Huang F X, Sun X M, Vasilakos A, Yang C N. Enabling semantic search based on conceptual graphs over encrypted outsourced data. *IEEE Trans. Services Computing*, 2016, doi: 10.1109/TSC.2016.2622697.

[4] Pan Z Q, Lei J J, Zhang Y, Sun X M, Kwong S. Fast motion estimation based on content property for low-complexity H.265/HEVC encoder. *IEEE Trans. Broadcasting*, 2016, 62(3): 675-684.

[5] Fu Z J, Ren K, Shu J G, Sun X M, Huang F X. Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Tran. Parallel and Distributed Systems*, 2016, 27(9): 2546-2559.

[6] Wen X Z, Shao L, Xue Y, Fang W. A rapid learning algorithm for vehicle classification. *Information Sciences*, 2015, 295: 395-406.

[7] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In *Proc IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.3431-3440.

[8] Lin G S, Shen C H, van den Hengel A, Reid I. Exploring context with deep structured models for semantic segmentation. *arXiv: 1603.03183*, 2017. https://arxiv.org/abs/1603.03183, June 2017.

[9] 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 *Proc. European Conf. Computer Vision*, October 2014, pp.740-755.

[10] Dai J F, He K M, Sun J. Instance-aware semantic segmentation via multi-task network cascades. In *Proc IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp.3150-3158.

[11] Pinheiro P H O, Collobert R, Dollár P. Learning to segment object candidates. In *Proc. the 28th Int. Conf. Neural Information Processing Systems*, December 2015, pp.1990-1998.

[12] Girshick R. Fast R-CNN. In *Proc IEEE Int. Conf. Computer Vision*, December 2015, pp.1440-1448.

[13] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137-1149.

[14] Dai J F, Li Y, He K M, Sun J. R-FCN: Object detection via region-based fully convolutional networks. In *Proc. the 30th Conf. Neural Information Processing Systems*, December 2016, pp.379-387.

[15] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In *Proc. the 25th Int. Conf. Neural Information Processing Systems*, December 2012, pp.1097-1105.

[16] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017, doi: 10.1109/TPAMI.2016.2644615.

[17] Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.1520-1528.

[18] Liu S, Wang C H, Qian R H, Yu H, Bao R D. Surveillance video parsing with single frame supervision. *arXiv: 1611.09587*, 2016. https://arxiv.org/abs/1611.09587, June 2017.

[19] Liu S, Liang X D, Liu L Q, Shen X H, Yang J C, Xu C S, Lin L, Cao X, Yan S C. Matching-CNN meets KNN: Quasi-parametric human parsing. In *Proc IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.1419-1427.

[20] Liu S, Liang X D, Liu L Q, Lu K, Lin L, Cao X C, Yan S C. Fashion parsing with video context. *IEEE Trans. Multimedia*, 2015, 17(8): 1347-1358.

[21] Liang X D, Liu S, Shen X H, Yang J C, Liu L Q, Dong J, Lin L, Yan S C. Deep human parsing with active template regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2015, 37(12): 2402-2414.

[22] Liu S, Ou X Y, Qian R H, Wang W, Cao X C. Makeup like a superstar: Deep localized makeup transfer network. In *Proc. the 25th Int. Joint Conf. Artificial Intelligence*, July 2016, pp.2568-2575.

[23] Liu S, Feng J S, Song Z, Zhang T Z, Lu H Q, Xu C S, Yan S C. Hi, magic closet, tell me what to wear! In *Proc. the 20th ACM Int. Conf. Multimedia*, October 2012, pp.619-628.

[24] Zhou B L, Khosla A, Lapedriza À, Torralba A, Oliva A. Places: An image database for deep scene understanding. *arXiv: 1610.02055*, 2016. https://arxiv.org/abs/1610.02055, June 2017.

[25] He K M, Zhang X Y, Ren S Q, Sun J. Identity mappings in deep residual networks. In *Proc. European Conf. Computer Vision*, October 2016, pp.630645.

[26] Dai J F, He K M, Sun J. Convolutional feature masking for joint object and stuff segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.3992-4000.

[27] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp.580-587.

[28] Hariharan B, Arbeláez P, Girshick R, Malik J. Simultaneous detection and segmentation. In *Proc. European Conf. Computer Vision*, October 2014, pp.297-312.

[29] Sharma A, Tuzel O, Liu M Y. Recursive context propagation network for semantic scene labeling. In *Proc. Annual Conf. Neural Information Processing Systems*, December 2014, pp.2447-2455.

[30] Sharma A, Tuzel O, Jacobs D W. Deep hierarchical parsing for semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.530-538.

[31] He K M, Zhang X Y, Ren S Q, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2015, 37(9): 1904-1916.

[32] Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z Z, Du D L, Huang C, Torr P H S. Conditional random fields as recurrent neural networks. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.1529-1537.

[33] Arnab A, Jayasumana S, Zheng S, Torr P H S. Higher order conditional random fields in deep neural networks. In *Proc. European Conf. Computer Vision*, October 2016, pp.524-540.

[34] Ciresan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.3642-3649.

[35] Dai J F, He K M, Sun J. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.1635-1643.

[36] Krähenbühl P, Koltun V. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. the 24th Int. Conf. Neural Information Processing Systems*, December 2011, pp.109-117.

[37] Jia Y Q, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R B, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. In *Proc. the 22nd ACM Int. Conf. Multimedia*, November 2014, pp.675-678.

[38] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. *arXiv: 1511.07122*, 2016. https://arxiv.org/abs/1511.07122, June 2017.

[39] 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp.3213-3223.

[40] Liu Z W, Li X X, Luo P, Loy C C, Tang X O. Semantic image segmentation via deep parsing network. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.1377-1385.

Xin-Yu Ou received his B.E. degree in electronic information science and technology and M.S. degree in software engineering from Yunnan University (YNU), Kunming, in 2004 and 2009 respectively. He is a Ph.D. candidate in computer science and technology of Huazhong University of Science and Technology (HUST), Wuhan, and is also a visiting Ph.D. in Institute of Information Engineering, Chinese Academy of Sciences (CASIIE), Beijing. He is an associate professor in Yunnan Open University (YNOU), Kunming. His research interests include deep learning, image retrieval, and object detection and recognition.

Ping Li is a lecturer in School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan. He received his Ph.D. degree in computer application from Huazhong University of Science and Technology, Wuhan, in 2009. His research interests include multimedia security and machine learning.

He-Fei Ling received his B.E. and M.S. degrees in energy and power engineering from Huazhong University of Science and Technology (HUST), Wuhan, in 1999 and 2002 respectively, and his Ph.D. degree in computer science from HUST, Wuhan, in 2005. From 2006 to 2011, he was an associate professor with the School of Computer Science and Technology, HUST, Wuhan. From 2008 to 2009, he joined in the Department of Computer Science, University College London (UCL) as a visiting scholar. Since 2011, he has been a full professor with the School of Computer Science and Technology, HUST, Wuhan. His research spans multimedia, intelligence, and security. Prof. Ling has coauthored over 100 publications in the leading scholarly journals in multimedia and security.

Si Liu is an associate professor in Institute of Information Engineering, Chinese Academy of Sciences, Beijing. She was a research fellow at Learning and Vision Group of National University of Singapore, Singapore. She received her Ph.D. degree in pattern recognition and intelligent system from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, in 2012. Her research interests include computer vision and multimedia.

Tian-Jiang Wang received his Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, in 2002. He has been a professor with the School of Computer Science and Technology of Huazhong University of Science and Technology, Wuhan. His research interest is in the artificial intelligence, machine learning, computer vision and virtual reality and he has published more than 50 papers in these research fields.

Dan Li received her B.E. and M.S. degrees in mechanical design, manufacturing and automation from Huazhong University of Science and Technology (HUST), Wuhan, in 1998 and 2002 respectively, and her Ph.D. degree in computer science from HUST, Wuhan, in 2008. Since 2015, she has been an associate professor with the School of Computer Science and Technology, HUST, Wuhan. Her research spans computer graphics, multimedia, and intelligence.

Appendix

A Performance of Each Category

The evaluation results of each category are list in Table A. The mIoU and accuracy indicate the pixel intersection-over-union averaged across the 150 classes and the average proportion of correctly classified pixel on each category, respectively. The bold indicates the best result in one category.

Table A. Performance of Each Category on SceneParse150 Validation Set

	Base	MMC	Box	Mask	BH		Base	MMC	Box	Mask	BH
Wall	66.0	69.4	68.9	**69.7**	69.4	Building	75.7	77.0	76.8	76.9	**77.7**
Sky	92.3	92.3	92.3	92.3	**92.3**	Floor	69.7	73.7	71.3	**73.8**	73.3
Tree	67.2	**69.2**	69.0	68.7	68.0	Ceiling	73.8	77.5	76.8	77.5	**77.7**
Road	73.8	77.5	76.8	77.5	**77.7**	Bed	73.9	**81.7**	72.9	81.3	81.2
Pane	48.7	**54.6**	52.2	53.5	53.4	Grass	**65.6**	62.8	63.4	62.9	63.6
Cabinet	47.9	50.8	**54.5**	52.9	53.0	Sidewalk	47.5	55.9	55.7	**55.9**	55.5
Person	67.4	70.7	66.5	**71.2**	70.6	Earth	27.8	22.9	22.5	22.8	**29.0**
Door	20.4	26.4	**34.1**	29.2	30.9	Table	38.3	**47.6**	42.3	46.9	46.0
Mountain	51.7	49.9	49.2	**49.5**	49.1	Plant	41.9	41.5	44.6	40.5	43.0
Curtain	60.1	65.6	64.1	**65.9**	65.6	Chair	36.7	44.0	45.1	45.8	**45.4**
Car	72.0	78.2	73.2	78.7	79.0	Water	45.4	47.1	46.9	47.1	**47.8**
Painting	57.4	63.8	62.9	**63.5**	63.2	Sofa	45.5	54.5	54.0	**54.9**	54.5
Shelf	28.6	34.6	34.7	34.5	**34.9**	House	43.9	34.7	34.7	34.8	**39.3**
Sea	46.2	52.2	51.6	52.1	**55.0**	Mirror	39.2	52.7	54.4	54.2	**54.6**
Rug	36.4	39.6	37.3	39.6	**39.7**	Field	**30.8**	23.8	22.7	22.9	21.3
Armchair	17.6	29.6	**36.1**	33.2	32.9	Seat	37.6	50.0	50.0	51.6	**53.0**
Fence	25.4	26.0	25.6	25.8	**28.0**	Desk	31.4	38.6	34.3	**39.9**	38.9
Rock	31.8	34.9	**37.5**	33.9	32.6	Wardrobe	39.7	42.2	43.0	44.6	**45.5**
Lamp	35.4	44.7	42.1	**44.5**	44.4	Bathtub	54.2	63.2	59.7	62.9	**63.2**
Railing	24.7	23.6	22.4	23.6	**24.8**	Cushion	26.2	38.0	31.1	38.3	**39.0**
Base	11.2	16.8	16.6	16.8	**18.8**	Box	9.1	7.0	10.8	9.8	**13.0**
Column	33.5	**38.9**	36.7	38.2	37.9	Signboard	22.0	23.3	22.8	23.3	**24.2**
Chest	36.8	41.6	46.0	48.1	**48.8**	Counter	28.3	27.9	27.6	**28.2**	27.9
Sand	18.9	28.1	28.7	28.1	**34.0**	Sink	44.1	52.7	53.1	57.1	**59.1**
Skyscraper	54.7	66.1	**66.2**	**66.2**	65.0	Fireplace	48.6	63.5	66.2	**64.4**	64.3
Refrigerator	43.3	67.1	64.7	**73.9**	73.2	Grandstand	30.4	31.0	29.3	31.0	**35.6**
Path	15.2	**20.6**	**20.6**	**20.6**	20.1	Stairs	**27.0**	22.5	22.4	22.5	21.9
Runway	**63.3**	57.1	57.1	57.2	62.4	Case	32.0	29.9	28.3	29.4	**35.0**
Pool Table	86.5	**88.4**	88.1	**88.4**	88.3	Pillow	34.2	**38.3**	21.0	37.5	37.2
Screen Door	30.0	35.1	30.3	32.8	**36.8**	Stairway	21.1	22.6	22.6	22.6	**22.7**
River	10.5	**14.7**	14.3	14.6	14.1	Bridge	18.1	23.4	23.4	23.5	**40.7**
Bookcase	30.7	27.6	**32.2**	30.0	29.9	Blind	13.1	14.9	**24.3**	19.9	19.9
Coffee Table	34.4	48.2	42.5	**48.8**	**48.8**	Toilet	63.8	77.7	73.2	**78.7**	77.9
Flower	20.2	25.6	24.7	25.2	**28.6**	Book	28.1	33.6	20.3	33.9	**36.6**
Hill	**6.3**	5.8	5.8	5.8	5.8	Bench	34.4	33.6	33.1	**34.5**	33.6
Countertop	39.2	42.1	40.3	44.5	**46.6**	Stove	47.4	**55.6**	49.3	51.0	52.6
Palm	**40.3**	35.9	31.3	32.6	32.0	Kitchen	29.0	24.5	27.6	**29.8**	29.2
Computer	44.9	54.9	45.9	**56.2**	55.6	Swivelchair	31.1	31.5	37.6	38.9	**40.0**
Boat	42.0	41.4	**57.1**	46.3	51.9	Bar	27.4	24.5	24.3	26.3	**27.8**
Arcade	27.7	25.5	31.1	25.6	**39.4**	Hovel	**21.4**	4.8	4.8	4.8	10.1
Bus	63.7	83.3	82.4	83.4	**84.4**	Towel	38.8	39.5	40.3	40.5	**41.4**
Light	15.4	**21.2**	**21.2**	**21.2**	**21.2**	Truck	8.5	21.1	**25.5**	24.5	24.0
Tower	26.1	32.9	**33.0**	32.9	32.7	Chandelier	43.9	50.5	48.3	54.9	**54.9**
Awning	10.4	12.5	**23.6**	20.8	20.6	Streetlight	3.6	**9.1**	7.5	8.9	9.0
Booth	30.6	37.7	37.4	37.9	**44.0**	Television	47.4	57.4	**62.4**	60.2	58.1
Airplane	50.4	52.1	46.7	**52.4**	46.6	Dirt Track	0.0	0.0	0.0	0.0	0.0
Apparel	20.0	17.7	14.5	17.1	**20.6**	Pole	4.0	5.5	**10.5**	8.8	8.9
Land	0.0	2.6	2.6	2.5	**5.4**	Bannister	4.0	2.0	2.0	2.0	**6.0**
Escalator	**32.1**	5.7	5.9	5.7	5.7	Ottoman	21.1	27.0	**31.2**	30.0	29.7
Bottle	2.1	16.2	20.0	18.2	**26.7**	Buffet	29.3	38.7	**40.9**	39.2	39.1
Poster	1.0	9.5	**21.7**	12.8	12.0	Stage	1.3	4.3	4.6	4.3	**6.4**
Van	23.3	29.8	21.8	42.0	**42.3**	Ship	26.7	4.3	18.9	4.6	**27.2**
Fountain	17.4	1.5	19.9	19.9	**19.7**	Conveyer	35.8	37.5	38.3	37.5	**50.3**
Canopy	4.7	13.8	**20.8**	14.4	14.2	Washer	50.0	36.0	34.0	35.9	**50.8**
Plaything	**18.6**	10.5	4.4	10.4	15.6	Swimpool	**18.0**	17.7	17.9	17.7	17.5

Table A. (Continued)

	Base	MMC	Box	Mask	BH		Base	MMC	Box	Mask	BH
Stool	13.8	26.0	**32.1**	28.0	29.2	Barrel	12.4	37.9	**41.2**	38.7	37.4
Basket	6.0	8.9	**18.1**	13.9	16.4	Waterfall	**67.1**	34.6	34.5	34.7	40.2
Tent	73.1	74.8	**74.9**	74.8	67.6	Bag	2.6	4.0	2.5	1.9	**6.3**
Minibike	27.6	54.4	42.5	52.2	**55.0**	Cradle	57.5	75.5	67.5	**75.6**	**75.6**
Oven	4.7	18.6	**19.2**	17.2	14.8	Ball	36.2	38.5	31.8	38.4	**40.4**
Food	22.4	4.7	17.8	15.7	**54.4**	Step	6.0	0.0	4.7	7.3	**8.0**
Tank	27.2	30.0	34.6	37.2	**40.2**	Trade	**14.0**	13.4	12.4	12.7	12.7
Microwave	28.7	43.2	44.0	44.7	**56.8**	Pot	19.6	13.4	11.0	13.4	**20.7**
Animal	29.4	29.2	33.3	33.1	**52.0**	Bicycle	34.1	37.7	**45.0**	38.9	40.5
Lake	2.4	**41.9**	**41.9**	**41.9**	41.4	Dishwasher	29.1	47.1	**49.0**	41.9	41.4
Screen	60.9	68.9	**69.8**	68.9	68.9	Blanket	0.0	2.6	0.0	4.1	**4.4**
Sculpture	11.2	32.3	34.5	34.0	**41.9**	Hood	23.1	35.0	24.4	35.1	**35.3**
Sconce	6.5	19.9	30.3	31.8	**31.9**	Vase	8.3	21.8	23.8	32.7	**33.2**
Traffic Light	9.5	18.5	**26.0**	20.0	19.9	Tray	0.0	2.1	3.6	4.0	**6.5**
Ashcan	9.3	25.4	22.3	24.8	**28.1**	Fan	**34.3**	30.3	29.2	30.2	30.0
Pier	**26.8**	12.7	12.7	12.7	12.7	Crt Screen	0.0	21.9	22.2	**23.0**	22.9
Plate	11.3	18.5	25.5	26.6	**39.5**	Monitor	6.5	5.0	13.7	18.3	**19.3**
Bulletin	29.2	32.0	**38.8**	32.3	29.4	Shower	0.0	1.0	**2.5**	2.0	2.0
Radiator	15.7	30.2	34.1	**43.3**	43.1	Glass	1.9	1.3	**5.0**	1.5	4.0
Clock	5.0	9.9	**16.7**	14.0	13.9	Flag	7.5	23.4	**30.8**	24.1	24.4
mIoU	30.9	35.5	35.6	36.5	38.4	Accuracy	74.0	75.5	75.2	75.7	77.9