Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited
{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

Abstract

Scene parsing is challenging for unrestricted open vocabulary and diverse scenes. In this paper, we exploit the capability of global context information by different-region-based context aggregation through our pyramid pooling module together with the proposed pyramid scene parsing network (PSPNet). Our global prior representation is effective to produce good quality results on the scene parsing task, while PSPNet provides a superior framework for pixel-level prediction. The proposed approach achieves state-of-the-art performance on various datasets. It came first in ImageNet scene parsing challenge 2016, PASCAL VOC 2012 benchmark and Cityscapes benchmark. A single PSPNet yields the new record of mIoU accuracy 85.4% on PASCAL VOC 2012 and accuracy 80.2% on Cityscapes.

(a) Image (b) Ground Truth

Figure 1. Illustration of complex scenes in ADE20K dataset.

1. Introduction

Scene parsing, based on semantic segmentation, is a fundamental topic in computer vision. The goal is to assign each pixel in the image a category label. Scene parsing provides complete understanding of the scene. It predicts the label, location, as well as shape for each element. This topic is of broad interest for potential applications of automatic driving, robot sensing, to name a few.

Difficulty of scene parsing is closely related to scene and label variety. The pioneer scene parsing task [23] is to classify 33 scenes for 2,688 images on LMO dataset [22]. More recent PASCAL VOC semantic segmentation and PASCAL context datasets [8, 29] include more labels with similar context, such as chair and sofa, horse and cow, etc. The new ADE20K dataset [43] is the most challenging one with a large and unrestricted open vocabulary and more scene classes. A few representative images are shown in Fig. 1. To develop an effective algorithm for these datasets needs to conquer a few difficulties.

State-of-the-art scene parsing frameworks are mostly based on the *fully convolutional network* (FCN) [26]. The deep *convolutional neural network* (CNN) based methods boost dynamic object understanding, and yet still face challenges considering diverse scenes and unrestricted vocabulary. One example is shown in the first row of Fig. 2, where a *boat* is mistaken as a *car*. These errors are due to similar appearance of objects. But when viewing the image regarding the context prior that the scene is described as *boathouse* near a river, correct prediction should be yielded.

Towards accurate scene perception, the knowledge graph relies on prior information of scene context. We found that the major issue for current FCN based models is lack of suitable strategy to utilize global scene category clues. For typical complex scene understanding, previously to get a global image-level feature, spatial pyramid pooling [18] was widely employed where spatial statistics provide a good descriptor for overall scene interpretation. Spatial pyramid pooling network [12] further enhances the ability.

Different from these methods, to incorporate suitable global features, we propose *pyramid scene parsing network* (PSPNet). In addition to traditional dilated FCN [3, 40] for pixel prediction, we extend the pixel-level feature to the specially designed global pyramid pooling one. The local and global clues together make the final prediction more reliable. We also propose an optimization strategy with

deeply supervised loss. We give all implementation details, which are key to our decent performance in this paper, and make the code and trained models publicly available [1].

Our approach achieves state-of-the-art performance on all available datasets. It is the champion of ImageNet scene parsing challenge 2016 [43], and arrived the 1st place on PASCAL VOC 2012 semantic segmentation benchmark [8], and the 1st place on urban scene Cityscapes data [6]. They manifest that PSPNet gives a promising direction for pixel-level prediction tasks, which may even benefit CNN-based stereo matching, optical flow, depth estimation, etc. in follow-up work. Our main contributions are threefold.

- We propose a pyramid scene parsing network to embed difficult scenery context features in an FCN based pixel prediction framework.
- We develop an effective optimization strategy for deep ResNet [13] based on deeply supervised loss.
- We build a practical system for state-of-the-art scene parsing and semantic segmentation where all crucial implementation details are included.

2. Related Work

In the following, we review recent advances in scene parsing and semantic segmentation tasks. Driven by powerful deep neural networks [17, 33, 34, 13], pixel-level prediction tasks like scene parsing and semantic segmentation achieve great progress inspired by replacing the fully-connected layer in classification with the convolution layer [26]. To enlarge the receptive field of neural networks, methods of [3, 40] used dilated convolution. Noh *et al*. [30] proposed a coarse-to-fine structure with deconvolution network to learn the segmentation mask. Our baseline network is FCN and dilated network [26, 3].

Other work mainly proceeds in two directions. One line [26, 3, 5, 39, 11] is with multi-scale feature ensembling. Since in deep networks, higher-layer feature contains more semantic meaning and less location information. Combining multi-scale features can improve the performance.

The other direction is based on structure prediction. The pioneer work [3] used conditional random field (CRF) as post processing to refine the segmentation result. Following methods [25, 41, 1] refined networks via end-to-end modeling. Both of the two directions ameliorate the localization ability of scene parsing where predicted semantic boundary fits objects. Yet there is still much room to exploit necessary information in complex scenes.

To make good use of global image-level priors for diverse scene understanding, methods of [18, 27] extracted global context information with traditional features not from deep neural networks. Similar improvement was made under object detection frameworks [35]. Liu *et al*. [24] proved that global average pooling with FCN can improve semantic segmentation results. However, our experiments show that these global descriptors are not representative enough for the challenging ADE20K data. Therefore, different from global pooling in [24], we exploit the capability of global context information by different-region-based context aggregation via our pyramid scene parsing network.

3. Pyramid Scene Parsing Network

We start with our observation and analysis of representative failure cases when applying FCN methods to scene parsing. They motivate proposal of our pyramid pooling module as the effective global context prior. Our *pyramid scene parsing network* (PSPNet) illustrated in Fig. 3 is then described to improve performance for open-vocabulary object and stuff identification in complex scene parsing.

3.1. Important Observations

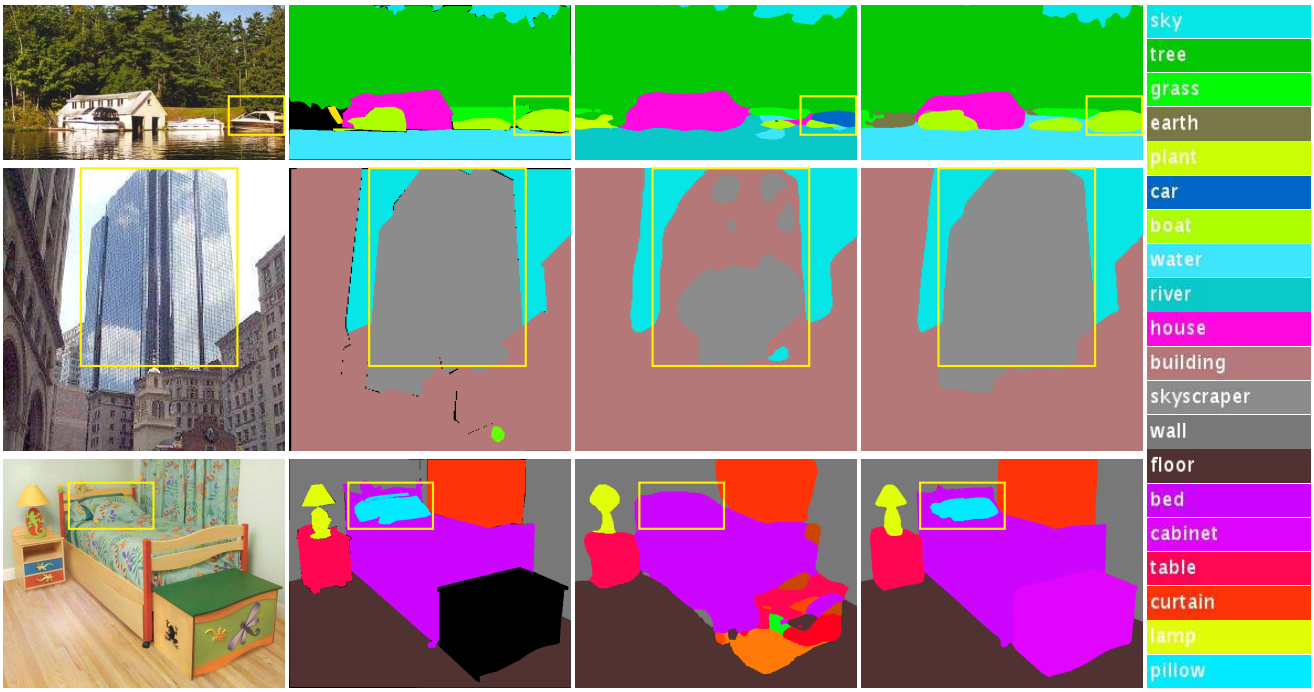
The new ADE20K dataset [43] contains 150 stuff/object category labels (*e.g*., wall, sky, and tree) and 1,038 image-level scene descriptors (*e.g*., airport_terminal, bedroom, and street). So a large amount of labels and vast distributions of scenes come into existence. Inspecting the prediction results of the FCN baseline provided in [43], we summarize several common issues for complex-scene parsing.

Mismatched Relationship Context relationship is universal and important especially for complex scene understanding. There exist co-occurrent visual patterns. For example, an airplane is likely to be in runway or fly in sky while not over a road. For the first-row example in Fig. 2, FCN predicts the boat in the yellow box as a "car" based on its appearance. But the common knowledge is that a car is seldom over a river. Lack of the ability to collect contextual information increases the chance of misclassification.

Confusion Categories There are many class label pairs in the ADE20K dataset [43] that are confusing in classification. Examples are *field* and *earth*; *mountain* and *hill*; *wall*, *house*, *building* and *skyscraper*. They are with similar appearance. The expert annotator who labeled the entire dataset, still makes 17.60% pixel error as described in [43]. In the second row of Fig. 2, FCN predicts the object in the box as part of *skyscraper* and part of *building*. These results should be excluded so that the whole object is either *skyscraper* or *building*, but not both. This problem can be remedied by utilizing the relationship between categories.

Inconspicuous Classes Scene contains objects/stuff of arbitrary size. Several small-size things, like streetlight and signboard, are hard to find while they may be of great importance. Contrarily, big objects or stuff may exceed the

[1] https://github.com/hszhao/PSPNet



(a) Image (b) Ground Truth (c) FCN (d) PSPNet (e) ColorMap

Figure 2. Scene parsing issues we observe on ADE20K [43] dataset. The first row shows the issue of mismatched relationship – cars are seldom over water than boats. The second row shows confusion categories where class "building" is easily confused as "skyscraper". The third row illustrates inconspicuous classes. In this example, the pillow is very similar to the bed sheet in terms of color and texture. These inconspicuous objects are easily misclassified by FCN.

receptive field of FCN and thus cause discontinuous prediction. As shown in the third row of Fig. 2, the pillow has similar appearance with the sheet. Overlooking the global scene category may fail to parse the pillow. To improve performance for remarkably small or large objects, one should pay much attention to different sub-regions that contain inconspicuous-category stuff.

To summarize these observations, many errors are partially or completely related to contextual relationship and global information for different receptive fields. Thus a deep network with a suitable global-scene-level prior can much improve the performance of scene parsing.

3.2. Pyramid Pooling Module

With above analysis, in what follows, we introduce the pyramid pooling module, which empirically proves to be an effective global contextual prior.

In a deep neural network, the size of receptive field can roughly indicates how much we use context information. Although theoretically the receptive field of ResNet [13] is already larger than the input image, it is shown by Zhou *et al*. [42] that the empirical receptive field of CNN is much smaller than the theoretical one especially on high-level layers. This makes many networks not sufficiently incorporate the momentous global scenery prior. We address this issue by proposing an effective global prior representation.

Global average pooling is a good baseline model as the global contextual prior, which is commonly used in image classification tasks [34, 13]. In [24], it was successfully applied to semantic segmentation. But regarding the complex-scene images in ADE20K [43], this strategy is not enough to cover necessary information. Pixels in these scene images are annotated regarding many stuff and objects. Directly fusing them to form a single vector may lose the spatial relation and cause ambiguity. Global context information along with sub-region context is helpful in this regard to distinguish among various categories. A more powerful representation could be fused information from different sub-regions with these receptive fields. Similar conclusion was drawn in classical work [18, 12] of scene/image classification.

In [12], feature maps in different levels generated by pyramid pooling were finally flattened and concatenated to be fed into a fully connected layer for classification. This global prior is designed to remove the fixed-size constraint of CNN for image classification. To further reduce context information loss between different sub-regions, we propose a hierarchical global prior, containing information with different scales and varying among different sub-regions. We

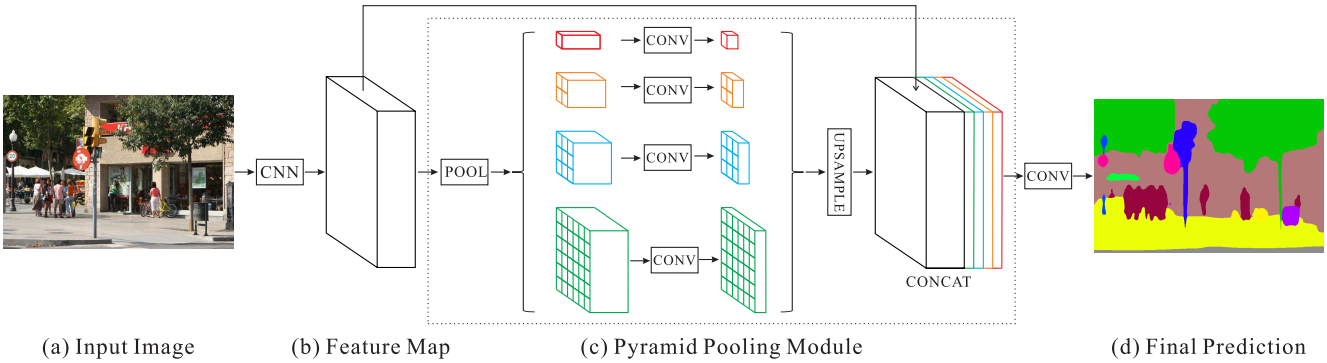


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

call it *pyramid pooling module* for global scene prior construction upon the final-layer-feature-map of the deep neural network, as illustrated in part (c) of Fig. 3.

The pyramid pooling module fuses features under four different pyramid scales. The coarsest level highlighted in red is global pooling to generate a single bin output. The following pyramid level separates the feature map into different sub-regions and forms pooled representation for different locations. The output of different levels in the pyramid pooling module contains the feature map with varied sizes. To maintain the weight of global feature, we use 1×1 convolution layer after each pyramid level to reduce the dimension of context representation to $1/N$ of the original one if the level size of pyramid is N. Then we directly upsample the low-dimension feature maps to get the same size feature as the original feature map via bilinear interpolation. Finally, different levels of features are concatenated as the final pyramid pooling global feature.

Noted that the number of pyramid levels and size of each level can be modified. They are related to the size of feature map that is fed into the pyramid pooling layer. The structure abstracts different sub-regions by adopting varying-size pooling kernels in a few strides. Thus the multi-stage kernels should maintain a reasonable gap in representation. Our pyramid pooling module is a four-level one with bin sizes of 1×1, 2×2, 3×3 and 6×6 respectively. For the type of pooling operation between max and average, we perform extensive experiments to show the difference in Section 5.2.

3.3. Network Architecture

With the pyramid pooling module, we propose our *pyramid scene parsing* network (PSPNet) as illustrated in Fig. 3. Given an input image in Fig. 3(a), we use a pretrained ResNet [13] model with the dilated network strategy [3, 40] to extract the feature map. The final feature map size is $1/8$ of the input image, as shown in Fig. 3(b). On top of the map, we use the pyramid pooling module shown in (c) to gather context information. Using our 4-level pyramid, the pooling kernels cover the whole, half of, and small portions of the image. They are fused as the global prior. Then we concatenate the prior with the original feature map in the final part of (c). It is followed by a convolution layer to generate the final prediction map in (d).

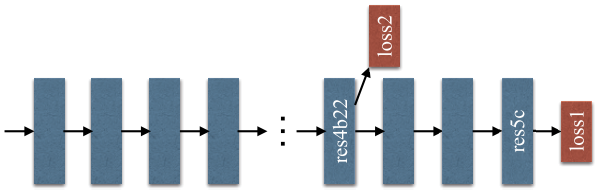


Figure 4. Illustration of auxiliary loss in ResNet101. Each blue box denotes a residue block. The auxiliary loss is added after the res4b22 residue block.

To explain our structure, PSPNet provides an effective global contextual prior for pixel-level scene parsing. The pyramid pooling module can collect levels of information, more representative than global pooling [24]. In terms of computational cost, our PSPNet does not much increase it compared to the original dilated FCN network. In end-to-end learning, the global pyramid pooling module and the local FCN feature can be optimized simultaneously.

4. Deep Supervision for ResNet-Based FCN

Deep pretrained networks lead to good performance [17, 33, 13]. However, increasing depth of the network may introduce additional optimization difficulty as shown in [32, 19] for image classification. ResNet solves this problem with skip connection in each block. Latter layers of deep ResNet mainly learn residues based on previous ones.

We contrarily propose generating initial results by supervision with an additional loss, and learning the residue afterwards with the final loss. Thus, optimization of the deep network is decomposed into two, each is simpler to solve.

An example of our deeply supervised ResNet101 [13] model is illustrated in Fig. 4. Apart from the main branch using softmax loss to train the final classifier, another classifier is applied after the fourth stage, i.e., the res4b22 residue block. Different from relay backpropagation [32] that blocks the backward auxiliary loss to several shallow layers, we let the two loss functions pass through all previous layers. The auxiliary loss helps optimize the learning process, while the master branch loss takes the most responsibility. We add weight to balance the auxiliary loss.

In the testing phase, we abandon this auxiliary branch and only use the well optimized master branch for final prediction. This kind of deeply supervised training strategy for ResNet-based FCN is broadly useful under different experimental settings and works with the pre-trained ResNet model. This manifests the generality of such a learning strategy. More details are provided in Section 5.2.

5. Experiments

Our proposed method is successful on scene parsing and semantic segmentation challenges. We evaluate it in this section on three different datasets, including ImageNet scene parsing challenge 2016 [43], PASCAL VOC 2012 semantic segmentation [8] and urban scene understanding dataset Cityscapes [6].

5.1. Implementation Details

For a practical deep learning system, devil is always in the details. Our implementation is based on the public platform Caffe [15]. Inspired by [4], we use the "poly" learning rate policy where current learning rate equals to the base one multiplying $(1 - \frac{iter}{max_iter})^{power}$. We set base learning rate to 0.01 and power to 0.9. The performance can be improved by increasing the iteration number, which is set to 150K for ImageNet experiment, 30K for PASCAL VOC and 90K for Cityscapes. Momentum and weight decay are set to 0.9 and 0.0001 respectively. For data augmentation, we adopt random mirror and random resize between 0.5 and 2 for all datasets, and additionally add random rotation between -10 and 10 degrees, and random Gaussian blur for ImageNet and PASCAL VOC. This comprehensive data augmentation scheme makes the network resist overfitting. Our network contains dilated convolution following [4].

During the course of experiments, we notice that an appropriately large "cropsize" can yield good performance and "batchsize" in the batch normalization [14] layer is of great importance. Due to limited physical memory on GPU cards, we set the "batchsize" to 16 during training. To achieve this, we modify Caffe from [37] together with branch [4] and make it support batch normalization on data gathered from multiple GPUs based on OpenMPI. For the auxiliary loss, we set the weight to 0.4 in experiments.

Method	Mean IoU(%)	Pixel Acc.(%)
ResNet50-Baseline	37.23	78.01
ResNet50+B1+MAX	39.94	79.46
ResNet50+B1+AVE	40.07	79.52
ResNet50+B1236+MAX	40.18	79.45
ResNet50+B1236+AVE	41.07	79.97
ResNet50+B1236+MAX+DR	40.87	79.61
ResNet50+B1236+AVE+DR	**41.68**	**80.04**

Table 1. Investigation of PSPNet with different settings. Baseline is ResNet50-based FCN with dilated network. 'B1' and 'B1236' denote pooled feature maps of bin sizes $\{1 \times 1\}$ and $\{1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6\}$ respectively. 'MAX' and 'AVE' represent max pooling and average pooling operations individually. 'DR' means that dimension reduction is taken after pooling. The results are tested on the validation set with the single-scale input.

5.2. ImageNet Scene Parsing Challenge 2016

Dataset and Evaluation Metrics The ADE20K dataset [43] is used in ImageNet scene parsing challenge 2016. Different from other datasets, ADE20K is more challenging for the up to 150 classes and diverse scenes with a total of 1,038 image-level labels. The challenge data is divided into 20K/2K/3K images for training, validation and testing. Also, it needs to parse both objects and stuff in the scene, which makes it more difficult than other datasets. For evaluation, both *pixel-wise accuracy* (Pixel Acc.) and *mean of class-wise intersection over union* (Mean IoU) are used.

Ablation Study for PSPNet To evaluate PSPNet, we conduct experiments with several settings, including pooling types of max and average, pooling with just one global feature or four-level features, with and without dimension reduction after the pooling operation and before concatenation. As listed in Table 1, average pooling works better than max pooling in all settings. Pooling with pyramid parsing outperforms that using global pooling. With dimension reduction, the performance is further enhanced. With our proposed PSPNet, the best setting yields results 41.68/80.04 in terms of Mean IoU and Pixel Acc. (%), exceeding global average pooling of 40.07/79.52 as idea in Liu *et al*. [24] by 1.61/0.52. And compared to the baseline, PSPNet outperforming it by 4.45/2.03 in terms of absolute improvement and 11.95/2.60 in terms of relative difference.

Ablation Study for Auxiliary Loss The introduced auxiliary loss helps optimize the learning process while not influencing learning in the master branch. We experiment with setting the auxiliary loss weight α between 0 and 1 and show the results in Table 2. The baseline uses ResNet50-based FCN with dilated network, with the master branch's softmax loss for optimization. Adding the auxiliary loss

Loss Weight α	Mean IoU(%)	Pixel Acc.(%)
ResNet50 (without AL)	35.82	77.07
ResNet50 (with $\alpha = 0.3$)	37.01	77.87
ResNet50 (with $\alpha = 0.4$)	**37.23**	**78.01**
ResNet50 (with $\alpha = 0.6$)	37.09	77.84
ResNet50 (with $\alpha = 0.9$)	36.99	77.87

Table 2. Setting an appropriate loss weight α in the auxiliary branch is important. 'AL' denotes the auxiliary loss. Baseline is ResNet50-based FCN with dilated network. Empirically, $\alpha = 0.4$ yields the best performance. The results are tested on the validation set with the single-scale input.

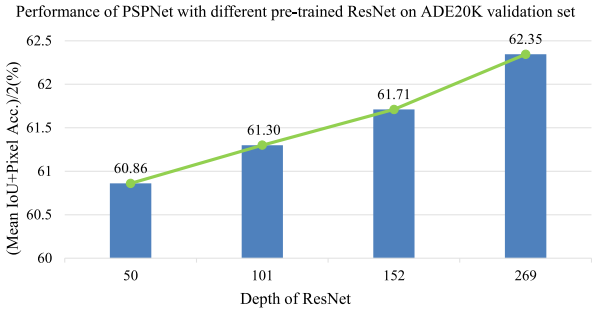


Figure 5. Performance grows with deeper networks. The results are obtained on the validation set with the single-scale input.

Method	Mean IoU(%)	Pixel Acc.(%)
PSPNet(50)	41.68	80.04
PSPNet(101)	41.96	80.64
PSPNet(152)	42.62	80.80
PSPNet(269)	**43.81**	**80.88**
PSPNet(50)+MS	42.78	80.76
PSPNet(101)+MS	43.29	81.39
PSPNet(152)+MS	43.51	81.38
PSPNet(269)+MS	**44.94**	**81.69**

Table 3. Deeper pre-trained model get higher performance. Number in the brackets refers to the depth of ResNet and 'MS' denotes multi-scale testing.

branch, $\alpha = 0.4$ yields the best performance. It outperforms the baseline with an improvement of 1.41/0.94 in terms of Mean IoU and Pixel Acc. (%). We believe deeper networks will benefit more given the new augmented auxiliary loss.

Ablation Study for Pre-trained Model Deeper neural networks have been shown in previous work to be beneficial to large scale data classification. To further analyze PSPNet, we conduct experiments for different depths of pre-trained ResNet. We test four depths of {50, 101, 152, 269}. As shown in Fig. 5, with the same setting, increasing the depth of ResNet from 50 to 269 can improve the score of (Mean IoU + Pixel Acc.) / 2 (%) from 60.86 to 62.35, with 1.49 absolute improvement. Detailed scores of PSPNet pre-trained from different depth ResNet models are listed in Table 3.

Method	Mean IoU(%)	Pixel Acc.(%)
FCN [26]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [40]	32.31	73.55
CascadeNet [43]	34.90	74.52
ResNet50-Baseline	34.28	76.35
ResNet50+DA	35.82	77.07
ResNet50+DA+AL	37.23	78.01
ResNet50+DA+AL+PSP	**41.68**	**80.04**
ResNet269+DA+AL+PSP	43.81	80.88
ResNet269+DA+AL+PSP+MS	**44.94**	**81.69**

Table 4. Detailed analysis of our proposed PSPNet with comparison with others. Our results are obtained on the validation set with the single-scale input except for the last row. Results of FCN, SegNet and DilatedNet are reported in [43]. 'DA' refers to data augmentation we performed, 'AL' denotes the auxiliary loss we added and 'PSP' represents the proposed PSPNet. 'MS' means that multi-scale testing is used.

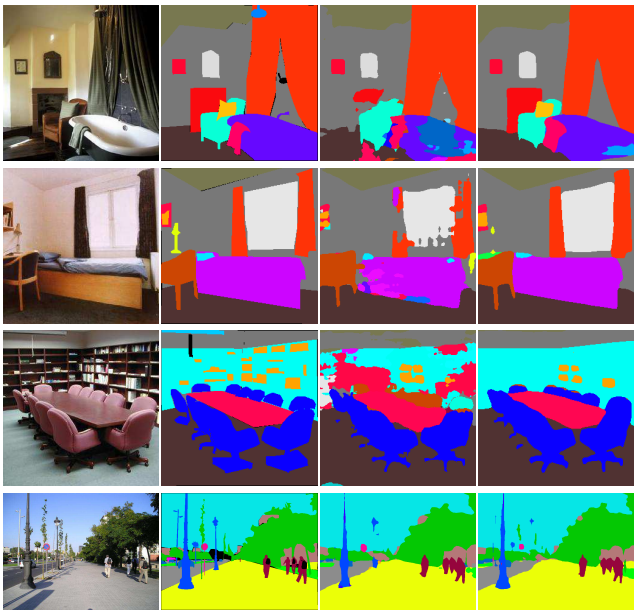
Rank	Team Name	Final Score (%)
1	**Ours**	**57.21**
2	Adelaide	56.74
3	360+MCG-ICT-CAS_SP	55.56
-	(our single model)	(55.38)
4	SegModel	54.65
5	CASIA_IVA	54.33
-	DilatedNet [40]	45.67
-	FCN [26]	44.80
-	SegNet [2]	40.79

Table 5. Results of ImageNet scene parsing challenge 2016. The best entry of each team is listed. The final score is the mean of Mean IoU and Pixel Acc. Results are evaluated on the testing set.

More Detailed Performance Analysis We show our more detailed analysis on the validation set of ADE20K in Table 4. All our results except the last-row one use single-scale test. "ResNet269+DA+AL+PSP+MS" uses multi-scale testing. Our baseline is adapted from ResNet50 with dilated network, which yields MeanIoU 34.28 and Pixel Acc. 76.35. It already outperforms other prior systems possibly due to the powerful ResNet [13].

Our proposed architecture makes further improvement compared to the baseline. Using data augmentation, our result exceeds the baseline by 1.54/0.72 and reaches 35.82/77.07. Using the auxiliary loss can further improve it by 1.41/0.94 and reaches 37.23/78.01. With PSPNet, we notice relatively more significant progress for improvement of 4.45/2.03. The result reaches 41.68/80.04. The difference from the baseline result is 7.40/3.69 in terms of absolute improvement and 21.59/4.83 (%) in terms of relativity. A deeper network of ResNet269 yields even higher performance up to 43.81/80.88. Finally, the multi-scale testing scheme moves the scores to 44.94/81.69.

Results in Challenge Using the proposed architecture, our team came in the 1st place in ImageNet scene parsing



(a) Image (b) Ground Truth (c) Baseline (d) PSPNet

Figure 6. Visual improvements on ADE20K, PSPNet produces more accurate and detailed results.

challenge 2016. Table 5 shows a few results in this competition. Our ensemble submission achieves score 57.21% on the testing set. Our single-model yields score 55.38%, which is even higher than a few other multi-model ensemble submissions. This score is lower than that on the validation set possibly due to the difference of data distributions between validation and testing sets. As shown in column (d) of Fig. 2, PSPNet solves the common problems in FCN. Fig. 6 shows another few parsing results on validation set of ADE20K. Our results contain more accurate and detailed structures compared to the baseline.

5.3. PASCAL VOC 2012

Our PSPNet also works satisfyingly on semantic segmentation. We carry out experiments on the PASCAL VOC 2012 segmentation dataset [8], which contains 20 object categories and one background class. Following the procedure of [26, 7, 31, 3], we use augmented data with the annotation of [10] resulting 10,582, 1,449 and 1,456 images for training, validation and testing. Results are shown in Table 6, we compare PSPNet with previous best-performing methods on the testing set based on two settings, i.e., with or without pre-training on MS-COCO dataset [21]. Methods pre-trained with MS-COCO are marked by '†'. For fair comparison with current ResNet based frameworks [38, 9, 4] in scene parsing/semantic segmentation task, we build our architecture based on ResNet101 while without post-processing like CRF. We evaluate PSPNet with several-scale input and use the average results following [3, 24].

(a) Image (b) Ground Truth (c) Baseline (d) PSPNet

Figure 7. Visual improvements on PASCAL VOC 2012 data. PSPNet produces more accurate and detailed results.

As shown in Table 6, PSPNet outperforms prior methods on both settings. Trained with only VOC 2012 data, we achieve 82.6% accuracy[2] – we get the highest accuracy on all 20 classes. When PSPNet is pre-trained with MS-COCO dataset, it reaches 85.4% accuracy[3] where 19 out of the 20 classes receive the highest accuracy. Intriguingly, our PSPNet trained with only VOC 2012 data outperforms existing methods trained with the MS-COCO pre-trained model.

One may argue that our based classification model is more powerful than several prior methods since ResNet was recently proposed. To exhibit our unique contribution, we show that our method also outperforms state-of-the-art frameworks that use the same model, including FCRNs [38], LRR [9], and DeepLab [4]. In this process, we even do not employ time-consuming but effective post-processing, such as CRF, as that in [4, 9].

Several examples are shown in Fig. 7. For "cows" in row one, our baseline model treats it as "horse" and "dog" while PSPNet corrects these errors. For "aeroplane" and "table" in the second and third rows, PSPNet finds missing parts. For "person", "bottle" and "plant" in following rows, PSPNet performs well on these small-size-object classes in the images compared to the baseline model. More visual comparisons between PSPNet and other methods are included in Fig. 9.

5.4. Cityscapes

Cityscapes [6] is a recently released dataset for semantic urban scene understanding. It contains 5,000 high quality pixel-level finely annotated images collected from 50 cities

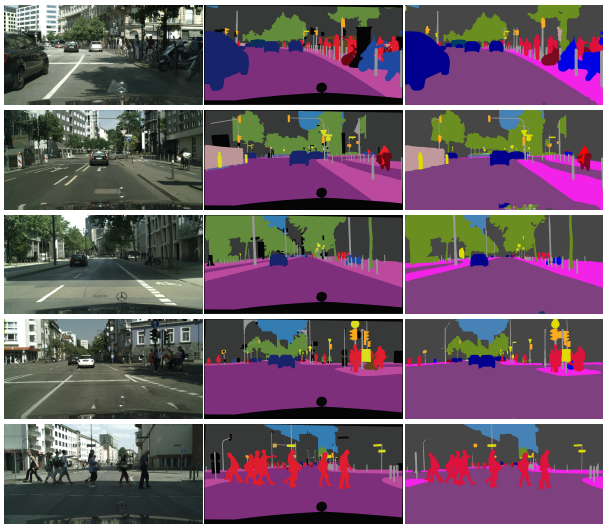
[2] http://host.robots.ox.ac.uk:8080/anonymous/0OOWLP.html
[3] http://host.robots.ox.ac.uk:8080/anonymous/6KIR41.html

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
FCN [26]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [28]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
DeepLab [3]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [41]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet [30]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
GCRF [36]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [25]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [20]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
PSPNet	**91.8**	**71.9**	**94.7**	**71.2**	**75.8**	**95.2**	**89.9**	**95.9**	**39.3**	**90.7**	**71.7**	**90.5**	**94.5**	**88.8**	**89.6**	**72.8**	**89.6**	**64.0**	**85.1**	**76.3**	**82.6**
CRF-RNN† [41]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup† [7]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
Dilation8† [40]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
DPN† [25]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Piecewise† [20]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
FCRNs† [38]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
LRR† [9]	92.4	45.1	94.6	65.2	75.8	**95.1**	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
DeepLab† [4]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
PSPNet†	**95.8**	**72.7**	**95.0**	**78.9**	**84.4**	94.7	**92.0**	**95.7**	**43.1**	**91.0**	**80.3**	**91.3**	**96.3**	**92.3**	**90.1**	**71.5**	**94.4**	**66.9**	**88.8**	**82.0**	**85.4**

Table 6. Per-class results on PASCAL VOC 2012 testing set. Methods pre-trained on MS-COCO are marked with '†'.

Method	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
CRF-RNN [41]	62.5	34.4	82.7	66.0
FCN [26]	65.3	41.7	85.7	70.1
SiCNN [16]	66.3	44.9	85.0	71.2
DPN [25]	66.8	39.1	86.0	69.1
Dilation10 [40]	67.1	42.0	86.5	71.1
LRR [9]	69.7	48.0	88.2	74.7
DeepLab [4]	70.4	42.6	86.4	67.7
Piecewise [20]	71.6	51.7	87.3	74.1
PSPNet	**78.4**	**56.7**	**90.6**	**78.6**
LRR‡ [9]	71.8	47.9	88.4	73.9
PSPNet‡	**80.2**	**58.1**	**90.6**	**78.2**

Table 7. Results on Cityscapes testing set. Methods trained using both fine and coarse data are marked with '‡'.



(a) Image (b) Ground Truth (c) PSPNet

Figure 8. Examples of PSPNet results on Cityscapes dataset.

in different seasons. The images are divided into sets with numbers 2,975, 500, and 1,525 for training, validation and testing. It defines 19 categories containing both stuff and objects. Also, 20,000 coarsely annotated images are provided for two settings in comparison, i.e., training with only fine data or with both the fine and coarse data. Methods trained using both fine and coarse data are marked with '‡'. Detailed results are listed in Table 7. Our base model is ResNet101 as in DeepLab [4] for fair comparison and the testing procedure follows Section 5.3.

Statistics in Table 7 show that PSPNet outperforms other methods with notable advantage. Using both fine and coarse data for training makes our method yield 80.2 accuracy. Several examples are shown in Fig. 8. Detailed per-class results on testing set are shown in Table 8.

6. Concluding Remarks

We have proposed an effective pyramid scene parsing network for complex scene understanding. The global pyramid pooling feature provides additional contextual information. We have also provided a deeply supervised optimization strategy for ResNet-based FCN network. We hope the implementation details publicly available can help the community adopt these useful strategies for scene parsing and semantic segmentation and advance related techniques.

Acknowledgements

We would like to thank Gang Sun and Tong Xiao for their help in training the basic classification models, Qun Luo for technical support. This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (project No. 2150760).

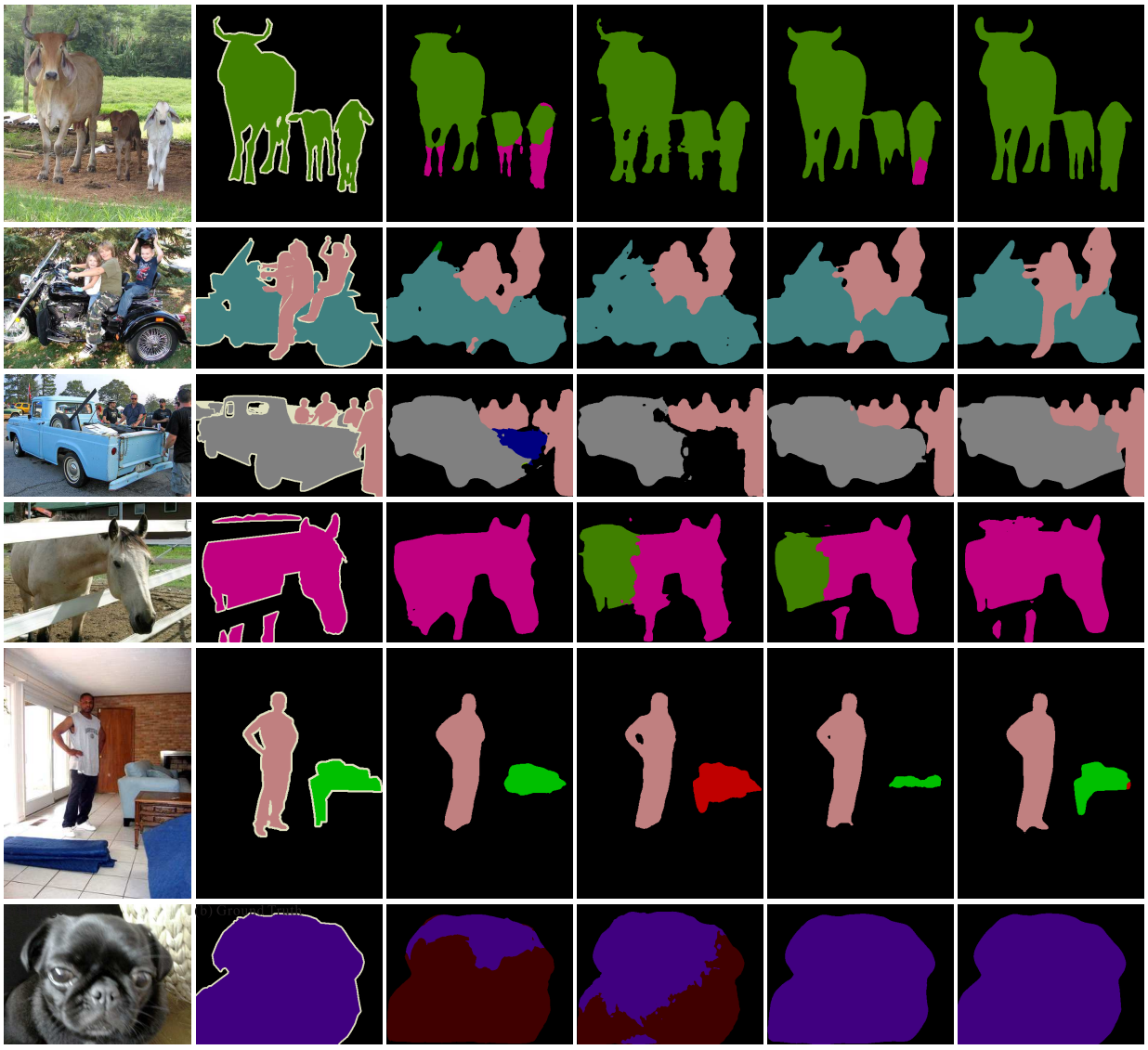


Figure 9. Visual comparison on PASCAL VOC 2012 data. (a) Image. (b) Ground Truth. (c) FCN [26]. (d) DPN [24]. (e) DeepLab [4]. (f) PSPNet.

Method	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
CRF-RNN [41]	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
FCN [26]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
SiCNN+CRF [16]	96.3	76.8	88.8	40.0	45.4	50.1	63.3	69.6	90.6	67.1	92.2	77.6	55.9	90.1	39.2	51.3	44.4	54.4	66.1	66.3
DPN [25]	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5	69.4	94.5	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8	66.8
Dilation10 [40]	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0	67.1
LRR [9]	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7	69.7
DeepLab [4]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
Piecewise [20]	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6	71.6
PSPNet	**98.6**	**86.2**	**92.9**	**50.8**	**58.8**	**64.0**	**75.6**	**79.0**	**93.4**	**72.3**	**95.4**	**86.5**	**71.3**	**95.9**	**68.2**	**79.5**	**73.8**	**69.5**	**77.2**	**78.4**
LRR‡ [9]	97.9	81.5	91.4	50.5	52.7	59.4	66.8	72.7	92.5	70.1	95.0	81.3	60.1	94.3	51.2	67.7	54.6	55.6	69.6	71.8
PSPNet‡	**98.6**	**86.6**	**93.2**	**58.1**	**63.0**	**64.5**	**75.2**	**79.2**	**93.4**	**72.1**	**95.1**	**86.3**	**71.4**	**96.0**	**73.5**	**90.4**	**80.3**	**69.9**	**76.9**	**80.2**

Table 8. Per-class results on Cityscapes testing set. Methods trained using both fine and coarse set are marked with '‡'.

References

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv:1511.00561*, 2015. 6

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv:1412.7062*, 2014. 1, 2, 4, 7, 8

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 5, 7, 8, 9

[5] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5, 7

[7] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 7, 8

[8] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes VOC challenge. *IJCV*, 2010. 1, 2, 5, 7

[9] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 7, 8, 9

[10] B. Hariharan, P. Arbelaez, L. D. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 7

[11] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1, 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 4, 5, 6

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 5

[16] I. Kreso, D. Causevic, J. Krapac, and S. Segvic. Convolutional scale invariance for semantic segmentation. In *GCPR*, 2016. 8, 9

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 4

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2, 3

[19] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 4

[20] G. Lin, C. Shen, I. D. Reid, and A. van den Hengel. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 8, 9

[21] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 7

[22] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009. 1

[23] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 2011. 1

[24] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv:1506.04579*, 2015. 2, 3, 4, 5, 7, 9

[25] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2, 8, 9

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 6, 7, 8, 9

[27] A. Lucchi, Y. Li, X. B. Bosch, K. Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? In *ICCV*, 2011. 2

[28] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015. 8

[29] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. L. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 1

[30] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2, 8

[31] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 7

[32] L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016. 4, 5

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 2, 4

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2, 3

[35] C. Szegedy, S. E. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv:1412.1441*, 2014. 2

[36] R. Vemulapalli, O. Tuzel, M. Liu, and R. Chellappa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, 2016. 8

[37] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv:1507.02159*, 2015. 5

[38] Z. Wu, C. Shen, and A. van den Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv:1605.06885*, 2016. 7, 8

[39] F. Xia, P. Wang, L. Chen, and A. L. Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *ECCV*, 2016. 2

[40] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv:1511.07122*, 2015. 1, 2, 4, 6, 8, 9

[41] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 8, 9

[42] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv:1412.6856*, 2014. 3

[43] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *arXiv:1608.05442*, 2016. 1, 2, 3, 5, 6