High Resolution Feature Recovering for Accelerating Urban Scene Parsing *

Rui Zhang[1,3], **Sheng Tang**[1], **Luoqi Liu**[2], **Yongdong Zhang**[1], **Jintao Li**[1], **Shuicheng Yan**[2,4]
[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] Qihoo 360 Artificial Intelligence Institute, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
zhangrui@ict.ac.cn; ts@ict.ac.cn; liuluoqi@360.cn; zhyd@ict.ac.cn; jtli@ict.ac.cn; yanshuicheng@360.cn.

Abstract

Both accuracy and speed are equally important in urban scene parsing. Most of the existing methods mainly focus on improving parsing accuracy, ignoring the problem of low inference speed due to large-sized input and high resolution feature maps. To tackle this issue, we propose a High Resolution Feature Recovering (HRFR) framework to accelerate a given parsing network. A Super-Resolution Recovering module is employed to recover features of large original-sized images from features of down-sampled input. Therefore, our framework can combine the advantages of (1) fast speed of networks with down-sampled input and (2) high accuracy of networks with large original-sized input. Additionally, we employ auxiliary intermediate supervision and boundary region re-weighting to facilitate the optimization of the network. Extensive experiments on the two challenging Cityscapes and CamVid datasets well demonstrate the effectiveness of the proposed HRFR framework, which can accelerate the scene parsing inference process by about 3.0× speedup from 1/2 down-sampled input with negligible accuracy reduction.

1 Introduction

Urban scene parsing is a significant and challenging task that benefits many applications, such as self-driving, driver assistance and traffic surveillance. The goal of urban scene parsing is to assign one of the semantic categories to each pixel within an urban scene image. Nowadays, approaches based on deep Convolutional Neural Networks (CNNs) [Krizhevsky *et al.*, 2012] achieve remarkable success in urban scene parsing, such as Fully Convolutional Network (FCN) based frameworks [Shelhamer *et al.*, 2017]. However, most of the existing scene parsing methods mainly focus on improving parsing accuracy through deeper networks [He *et al.*, 2016] and higher resolution feature maps [Chen *et al.*, 2016a] which result in lower inference speed. In practical applications, it is worth noting that both accuracy and speed are equally important in urban scene parsing.

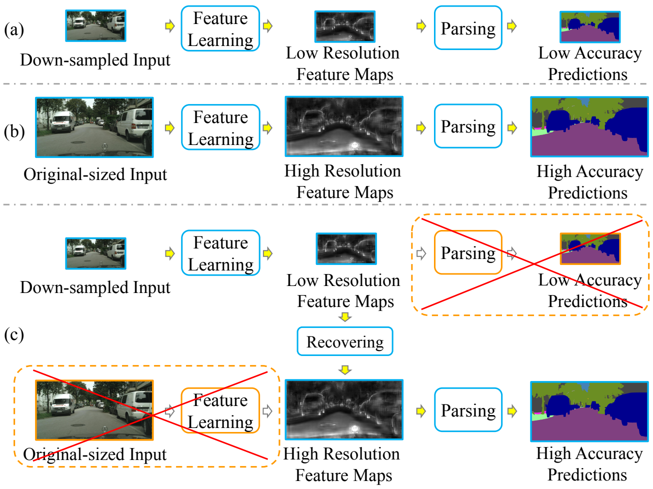

Figure 1: (a) Fast parsing with down-sampled input at the cost of low parsing accuracy. (b) High accuracy with large original-sized input at the cost of low inference speed. (c) The proposed HRFR combines the advantages of these two cases along the blue branch (with the orange branches abandoned).

Recently, most of the network accelerating approaches are introduced to image classification task, such as parameter approximation [Yang *et al.*, 2015; Wu *et al.*, 2016a] or network mimicking [Hinton *et al.*, 2015; Romero *et al.*, 2015]. Nevertheless, there is a huge difference between classification and scene parsing. For image classification, the size of input images is small (*e.g.* 224×224) and the resolution of feature maps are low (*e.g.*1/32 of input size). On the contrary, scene parsing task usually has large-sized input images (*e.g.* 1024×2048 in Cityscapes dataset) and high resolution feature maps (*e.g.*1/8 of input size) to capture more details. Both of these two characteristics lead to low inference speed but are not encountered in classification tasks. To tackle this issue, many approaches down-sample the input images [Paszke *et al.*, 2016; Badrinarayanan *et al.*, 2017], which speeds up the inference process but sacrifices the parsing accuracy due to missing details during down-sampling.

Aiming at the problem of low inference speed due to large-sized input and high resolution feature maps in scene parsing,

*Corresponding author: Sheng Tang. This work was supported by the National Key Research and Development Program of China (2017YFC0820605), and the National Natural Science Foundation of China (61525206, 61572472).

in this paper, we propose a High Resolution Feature Recovering (HRFR) framework to accelerate the inference process of a given scene parsing network. It combines the advantages of the following two cases while trying to avoid both disadvantages: (1) fast parsing with down-sampled input at the cost of low parsing accuracy as shown in Figure 1 (a); (2) high accuracy with large original-sized input at the cost of low inference speed as shown in Figure 1 (b). Briefly speaking, the core idea of our HRFR is to recover features of large original-sized images from features of down-sampled input, as shown in Figure 1 (c). High efficiency can be achieved since learning features from the down-sampled input is much faster than that from the original-sized input. Moreover, recovering high resolution features of original-sized input can provide more information about details and reduce the dropping of parsing accuracy. This is feasible because the features learned from the above two cases are overall similar in spite of blur boundaries and missing details in feature maps of down-sampled input, as visualized in Figure 2.

Particularly, we apply a Super-Resolution Recovering (SRR) module to perform feature recovering. The SRR module consists of several convolutional/deconvolutional layers with a bottle-neck structure, and can be jointly learned with the whole framework end-to-end. Additionally, we employ auxiliary intermediate supervision and boundary region re-weighting to enhance the optimization of the network.

Our proposed framework is oriented toward urban scene parsing by focusing on the problem of large-sized input and high resolution feature maps, wholly different with classification-based network accelerating approaches. The proposed method is orthogonal to most of the existing network accelerating methods and can be combined with them for further acceleration. Experimental results on the challenging Cityscapes dataset [Cordts *et al.*, 2016] and CamVid dataset [Brostow *et al.*, 2009] show the effectiveness of our proposed method, which can accelerate the scene parsing inference process by about 3.0× speedup from 1/2 down-sampled input with negligible accuracy reduction.

2 Related Work

Scene parsing Compared with graph-based methods [Xie *et al.*, 2014], effective and efficient approaches based on CNNs [Krizhevsky *et al.*, 2012] achieve extraordinary success in scene parsing task, such as FCN based frameworks [Shelhamer *et al.*, 2017; Chen *et al.*, 2016a] and Deconvolutional Network based frameworks [Badrinarayanan *et al.*, 2017]. Most of them aim at improving the parsing accuracy. Some approaches obtain higher resolution feature maps through atrous convolutions [Chen *et al.*, 2016a] or full-resolution residual architecture [Pohlen *et al.*, 2017]. Some approaches exploit features of multiple scales from intermediate layers [Mostajabi *et al.*, 2015; Ghiasi and Fowlkes, 2016], spatial pyramid structures [Zhao *et al.*, 2017; Chen *et al.*, 2016a], multiple-sized input [Chen *et al.*, 2016b] or adaptive learning [Zhang *et al.*, 2017b]. Some approaches utilize contextual information and spatial dependencies through Recurrent Neural Networks (RNNs) [Shuai *et al.*, 2016; Liang *et al.*, 2016] or graphical models [Liu *et al.*, 2015; Vemulapalli *et al.*, 2016; Lin *et al.*, 2016]. Other approaches focus on refinement to boost the accuracy of existing parsing predictions [Yu and Koltun, 2016; Zhang *et al.*, 2017a]. However, most of these approaches only consider the accuracy but ignore the inference speed. To boost the parsing accuracy, they utilize complicated structures which are slow and maybe not suitable for self-driving and traffic surveillance.

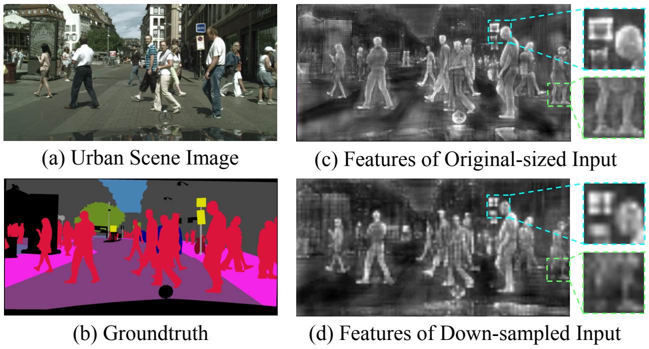
(a) Urban Scene Image
(c) Features of Original-sized Input
(b) Groundtruth
(d) Features of Down-sampled Input

Figure 2: Although features learned from the original-sized input and down-sampled input have similar responses, features learned from the original-sized input (c) have more details and more distinct boundaries than that from the down-sampled input (d).

Network Acceleration Network acceleration has received considerable attention in the recent years. Some approaches investigate approximation-based methods to reduce the redundancy in parameters and accelerate networks through low-rank decompositions of filters [Denton *et al.*, 2014; Jaderberg *et al.*, 2014; Yang *et al.*, 2015], product quantization of network parameters [Han *et al.*, 2016; Wu *et al.*, 2016a] or approximate networks by binary weights [Soudry *et al.*, 2014; Courbariaux *et al.*, 2015]. Some researchers improve the efficiency of networks by distilling the knowledge from multiple cumbersome models [Hinton *et al.*, 2015]. Network mimicking methods extend the distillation approach by training a student network with fewer parameters from mimicking a teacher network [Romero *et al.*, 2015; Ba and Caruana, 2014; Iandola *et al.*, 2016; Li *et al.*, 2017], so as to reduce the computation cost while preserving the accuracy. However, most of these model accelerating approaches focus on speedup models of image classification. They consider the redundancy of network structure instead of high resolution of feature maps, which remains a problem leading to slow inference speed and high memory cost in scene parsing.

In this paper, we propose the HRFR framework to speed up a given scene parsing network. In contrast to the existing network accelerating methods which focus on network structures of image classification, the proposed method aims at solving the problem of large-sized input and high resolution feature maps in scene parsing.

3 High Resolution Feature Recovering

In this section, we first introduce the overall structure of the proposed HRFR framework, as shown in Figure 3, followed by the description of the important SRR module, auxiliary intermediate supervision and boundary region re-weighting. Finally, we provide the complexity analysis.

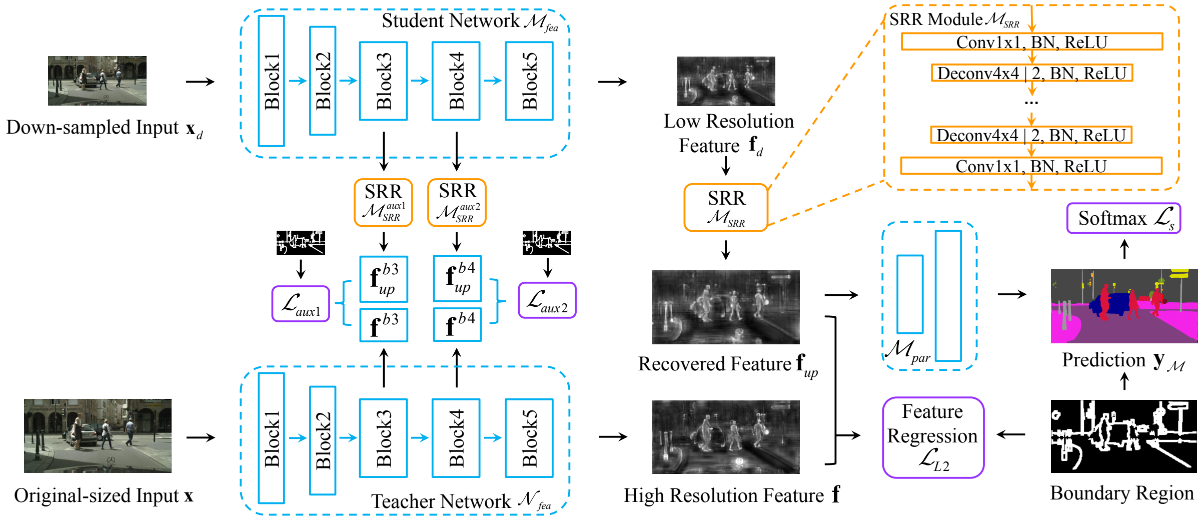
Student Network $\mathcal{M}_{fea}$
Block1
Block2
Block3
Block4
Block5
Down-sampled Input $\mathbf{x}_d$
Low Resolution Feature $\mathbf{f}_d$
SRR Module $\mathcal{M}_{SRR}$
Conv1x1, BN, ReLU
Deconv4x4 | 2, BN, ReLU
...
Deconv4x4 | 2, BN, ReLU
Conv1x1, BN, ReLU
SRR $\mathcal{M}_{SRR}^{aux1}$
SRR $\mathcal{M}_{SRR}^{aux2}$
SRR $\mathcal{M}_{SRR}$
$\mathcal{L}_{aux1}$
$\mathbf{f}_{up}^{b3}$
$\mathbf{f}^{b3}$
$\mathbf{f}_{up}^{b4}$
$\mathbf{f}^{b4}$
$\mathcal{L}_{aux2}$
Softmax $\mathcal{L}_s$
Recovered Feature $\mathbf{f}_{up}$
$\mathcal{M}_{par}$
Prediction $\mathbf{y}_{\mathcal{M}}$
Feature Regression $\mathcal{L}_{L2}$
Original-sized Input $\mathbf{x}$
Teacher Network $\mathcal{N}_{fea}$
High Resolution Feature $\mathbf{f}$
Boundary Region

Figure 3: The framework of the proposed HRFR method. The student network is composed of three subnets and is guided by the teacher network. The SRR module contains several deconvolutional layers with a bottle-neck structure.

3.1 Overall Framework

Given an urban scene image $\mathbf{x}$ (with the original size), the goal of urban scene parsing is to output the pixel-level prediction $\mathbf{y}_{\mathcal{N}}$ through a feed-forward convolutional neural network $\mathcal{N}$, formulated as

$$\mathbf{y}_{\mathcal{N}} = \mathcal{N}(\mathbf{x}). \tag{1}$$

Most of the popular scene parsing frameworks are based on the FCN framework, which can be decomposed into two consecutive subnets, feature learning subnet and parsing subnet. The feature learning subnet often contains dozens or hundreds of convolutional layers transferred from classification networks, while the parsing subnet only contains several layers initialized by random values. Thus, we decompose $\mathcal{N}$ into the feature learning subnet $\mathcal{N}_{fea}$ and parsing subnet $\mathcal{N}_{par}$, thus Equation (1) is decomposed to

$$\mathbf{f} = \mathcal{N}_{fea}(\mathbf{x}), \mathbf{y}_{\mathcal{N}} = \mathcal{N}_{par}(\mathbf{f}), \tag{2}$$

where $\mathbf{f}$ represents the feature maps learned from the feature learning subnet. These feature maps encode rich semantic concepts and spatial location information from the urban scene image $\mathbf{x}$. We name model $\mathcal{N}$ as the "teacher network", which represents the network trained on the original-sized images to learn feature maps of a high resolution.

The goal of the proposed HRFR method is to learn a "student network" $\mathcal{M}$ with down-sampled scene images to recover the high resolution $\mathbf{f}$ learned from the teacher network $\mathcal{N}$. We set the student network to the same architecture with the teacher network for simplification, but the input sizes of them are different. The feature maps $\mathbf{f}_d$ learned from $\mathcal{M}$ have a low resolution, formulated as

$$\mathbf{f}_d = \mathcal{M}_{fea}(\mathbf{x}_d), \tag{3}$$

where $\mathbf{x}_d$ denotes the image down-sampled from $\mathbf{x}$, $\mathcal{M}_{fea}$ denotes the feature learning subnet of $\mathcal{M}$. To recover the high resolution feature maps $\mathbf{f}$ learned from $\mathcal{N}$, we design a SRR module $\mathcal{M}_{SRR}$, which recovers the low resolution $\mathbf{f}_d$ to the high resolution $\mathbf{f}_{up}$ guided by $\mathbf{f}$, formulated as

$$\mathbf{f}_{up} = \mathcal{M}_{SRR}(\mathbf{f}_d). \tag{4}$$

Finally, the parsing prediction $\mathbf{y}_{\mathcal{M}}$ of $\mathcal{M}$ is obtained from the recovered high resolution feature maps $\mathbf{f}_{up}$:

$$\mathbf{y}_{\mathcal{M}} = \mathcal{M}_{par}(\mathbf{f}_{up}). \tag{5}$$

In the training stage, we aim at optimizing the student network $\mathcal{M}$ (including $\mathcal{M}fea$, $\mathcal{M}_{SRR}$ and $\mathcal{M}_{par}$) guided by the trained teacher network $\mathcal{N}$. The recovered feature maps $\mathbf{f}_{up}$ is expected to approximate the high resolution $\mathbf{f}$ learned from $\mathcal{N}$. Therefore, we use the L2 distance $\mathcal{L}_{L2}$ to minimize the differences between $\mathbf{f}_{up}$ and $\mathbf{f}$:

$$\mathcal{L}_{L2} = ||\mathbf{f}_{up} - \mathbf{f}||_2^2. \tag{6}$$

To further optimize the parsing subnet $\mathcal{M}_{par}$, the original softmax loss $\mathcal{L}_s$ between parsing prediction $\mathbf{y}_{\mathcal{M}}$ and groundtruth should be preserved. Thus the overall objective function $\mathcal{L}$ is defined as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{L2} + \lambda_2 \mathcal{L}_s, \tag{7}$$

where λ_1 and λ_2 are the loss weights. The feature maps $\mathbf{f}$ learned from parsing networks are often high-dimensional and difficult to be regressed. To provide a good starting state for the optimization process, we initialize $\mathcal{M}_{fea}$ and $\mathcal{M}_{par}$ in the student network with the parameters from $\mathcal{N}_{fea}$ and $\mathcal{N}_{par}$ in the teacher network instead of that from the classification network. This is because the teacher network can capture the semantic concepts of scenes better than image classification networks. What's more, feature maps generated from original-sized input and down-sampled input are similar, which can also reduce the difficulty of feature recovering. Additionally, parameters of $\mathcal{M}_{SRR}$ are initialized with random values sampled from the Gaussian distribution. In the inference stage, we only use the student network and the down-sampled scene image to gain the parsing prediction. The teacher network is only exploited to optimize the student network during training.

3.2 Super-Resolution Recovering

The SRR module $\mathcal{M}_{SRR}$ is designed for generating high resolution $\mathbf{f}_{up}$ from the low resolution $\mathbf{f}_d$. For convenience, we suppose the input scene images are down-sampled by a factor of $1/2^n, n = 1, 2, \cdots$. Thus the size of resultant low resolution feature maps $\mathbf{f}_d$ is correspondingly reduced to $1/2^n$ times of high resolution $\mathbf{f}$. To recover the responses of $\mathbf{f}$, we enlarge $\mathbf{f}_d$ to the same size of $\mathbf{f}$ by exploiting n deconvolutional layers with kernel= 4, stride= 2, each of which can magnify the feature maps by a factor of 2. The structure of several deconvolutional layers is compatible with the whole network structure, so that all the components can be conveniently integrated for joint training in an end-to-end manner.

It is worth noting that the channel numbers of feature maps $\mathbf{f}_{up}$ and $\mathbf{f}$ are usually very large, *e.g.* 2048 for ResNet. Thus each deconvolutional layer connects the two high-dimensional ends and has a large number of parameters of $4 \times 4 \times c \times c$, where c is the channel number and 4 is the kernel size. These huge number of parameters are redundant, causing slow inference speed. To solve this problem, we modify the structure of SRR module with a bottle-neck design inspired by [He *et al.*, 2016]. We add 1×1 convolutional layers at the beginning and ending of SRR module for reducing and restoring dimensions respectively. Suppose 1×1 convolutional layers reduce the dimension from c to $k(k < c)$, the parameter numbers of SRR module will decrease from $n \times 4^2 \times c^2$ to $n \times 4^2 \times k^2 + 2 \times c \times k$. For example, if we set $k = c/4$, the bottle-neck structure will remove more than 90% parameters compared with the original structure. Experiments also show that the bottle-neck design will speed up the inference process without damaging the parsing accuracy.

3.3 Auxiliary Intermediate Supervision

Deeper CNNs can bring better accuracy but lead to additional optimization difficulties, which can also be exhibited in the proposed HRFR framework. To conquer this problem, we exploit two auxiliary regression losses. Intuitively, if the feature maps from previous layers can be recovered, the feature maps of latter layers will be easier to be recovered. Therefore, we apply another two SRR modules after the 3rd and 4th blocks of the networks (*e.g.* the res3b3 and res4b22 of ResNet-101) to recover the intermediate feature maps and collect the regression losses, formulated as

$$\mathbf{f}_{up}^{b3} = \mathcal{M}_{SRR}^{aux1}(\mathbf{f}_d^{b3}), \mathbf{f}_{up}^{b4} = \mathcal{M}_{SRR}^{aux2}(\mathbf{f}_d^{b4}), \tag{8}$$

$$\mathcal{L}_{aux1} = ||\mathbf{f}_{up}^{b3} - \mathbf{f}^{b3}||_2^2, \mathcal{L}_{aux2} = ||\mathbf{f}_{up}^{b4} - \mathbf{f}^{b4}||_2^2, \tag{9}$$

where $\mathbf{f}_d^{b3}, \mathbf{f}_d^{b4}$ are the intermediate feature maps learned from the 3rd and 4th blocks of the student network, $\mathbf{f}_{up}^{b3}, \mathbf{f}_{up}^{b4}$ are the feature maps produced from the auxiliary SRR modules $\mathcal{M}_{SRR}^{aux1}, \mathcal{M}_{SRR}^{aux2}$. $\mathbf{f}^{b3}, \mathbf{f}^{b4}$ are the feature maps learned from the 3rd and 4th blocks of the teacher network, which are employed as the target of the auxiliary regression losses $\mathcal{L}_{aux1}$ and $\mathcal{L}_{aux2}$. Thus the objective function in Equation (7) changes to

$$\mathcal{L} = \lambda_1 \mathcal{L}_{L2} + \lambda_2 \mathcal{L}_s + \lambda_3(\mathcal{L}_{aux1} + \mathcal{L}_{aux2}). \tag{10}$$

We balance the auxiliary losses $\mathcal{L}_{aux1} + \mathcal{L}_{aux2}$ with smaller weights λ_3, so that the auxiliary losses can contribute to optimize the learning process while the loss of master regression branch $\mathcal{L}_{L2}$ and the softmax loss $\mathcal{L}_s$ take the most responsibility. With the help of auxiliary regression losses, the optimization of feature recovering is decomposed into three implicit stages, each is easier to optimize. Moreover, all the components of the network can be trained in balance instead of paying too much attention in the last subnet, which can also boost the parsing accuracy. During the inference stage, we simply abandon the two auxiliary branches and only recover the feature maps from the master branch.

3.4 Boundary Region Re-weighting

As visualized in Figure 2, the feature maps from the original-sized input resemble those from down-sampled input in overall. However, there are still some obvious distinctions such as blur details and inaccurate boundaries in down-sampled feature maps, which is the principal reason of accuracy degradation. We name the pixels or feature vectors near boundaries as boundary region. Thus, paying more attention to the optimization of the boundary regions will benefit feature recovering. We implement it by re-weighting the loss functions separately to increase the loss weights of predictions within boundary regions. Particularly, we obtain the boundary regions B through expanding the boundaries with a radius r. Then the objective functions are modified to

$$\mathcal{L}_{re} = \gamma_1 \sum \mathcal{L}(p|p \in B) + \gamma_2 \sum \mathcal{L}(p|p \notin B), \tag{11}$$

where $\mathcal{L}(p)$ represents the overall loss $\mathcal{L}$ defined in Equation (10) from the pixel p. We set $\gamma_1 > \gamma_2$ to pay more attention to optimizing the boundary regions.

3.5 Complexity Analysis

The computational cost ratio of the proposed method to the original method (i.e. the teacher network) can be formulated with the complexities of all the subnets

$$r = \frac{O(\mathcal{M}_{fea}) + O(\mathcal{M}_{SPR}) + O(\mathcal{M}_{par})}{O(\mathcal{N}_{fea}) + O(\mathcal{N}_{par})}, \tag{12}$$

where $O(\cdot)$ measures the function complexity. Since the feature learning subnet often contains much more fully convolutional layers than those of the parsing subnet, we have $O(\mathcal{N}_{par}) \ll O(\mathcal{N}_{fea})$ and $O(\mathcal{M}_{par}) \ll O(\mathcal{M}_{fea})$. Similarly, the SRR module only contains several convolutional/deconvolutional layers, so we also have $O(\mathcal{M}_{SRR}) \ll O(\mathcal{M}_{fea})$. Considering the teacher and student networks have the same architecture but different sizes of input, the complexity cost ratio is mostly determined by the down-sampling factor. The speedup ratio is approximately in inverse proportion to the computational cost ratio and can be estimated with the FLOPs of the student and teacher networks.

4 Experiments

In this section, we perform experiments on the two challenging urban scene parsing benchmarks, including Cityscapes dataset and CamVid dataset.

4.1 Experimental Settings

Cityscapes Dataset The Cityscapes dataset is taken by car-carried cameras and collected in street scenes from 50 different cities. It contains 5000 images of 19 semantic classes, 2975 images for training, 500 images for validation and 1525

Method	Mean IoU(%)	Time(ms)
1-DeepLab-v2	74.33	2366.4
1/2-DeepLab-v2	69.82	652.9
1/2-HRFR(Non-neck)	71.87	869.5
1/2-HRFR(Neck)	71.86	778.6
1/2-HRFR(Neck+Aux)	72.79	778.6
1/2-HRFR(Neck+Bound)	72.90	778.6
1/2-HRFR(Neck+Aux+Bound)	73.61	778.6

Table 1: Accuracy and inference time of the proposed method, evaluated on Cityscapes validation set on the DeepLab-v2 framework. 1 (1/2)-DeepLab-v2: DeepLab-v2 results with original-sized input (1/2 down-sampled input). Neck (Non-neck): the SRR module with (without) bottle-neck structure. Aux: applying auxiliary SRR modules. Bound: applying boundary region re-weighting.

Method	Mean IoU(%)	Time(ms)
1-DeepLab-v2	74.33	2366.4
1/2-DeepLab-v2	69.82	652.9
1/4-DeepLab-v2	60.35	182.3
1/8-DeepLab-v2	45.02	73.2
1/2-HRFR	73.61	778.6
1/4-HRFR	68.94	347.5
1/8-HRFR	59.73	252.8

Table 2: Accuracy and inference time of the proposed method with different down-sampling factors, evaluated on Cityscapes validation set with the DeepLab-v2 framework. 1 (1/2, 1/4 or 1/8)-DeepLab-v2: the original-sized input (1/2, 1/4 or 1/8 down-sampled input) for DeepLab-v2. 1/2 (1/4 or 1/8)-HRFR: the 1/2 (1/4 or 1/8) down-sampled input for the proposed HRFR method.

for testing. The Intersection over Union averaged over all the semantic categories (Mean IoU) is adopted for evaluation.

CamVid Dataset The CamVid dataset is collected with images captured from driving videos at daytime and dusk. It contains 701 images with pixel-level annotations on 11 semantic classes. The performance is evaluated based on the average per-pixel accuracy over all the semantic categories (CA) and the overall pixel-wise accuracy (PA).

It is worth noting that images in these two datasets have high resolutions, 2048×1024 of Cityscapes and 960×720 of CamVid. Many scene parsing approaches down-sample images of the two datasets to speed up inference, which damages the parsing accuracy. Thus, it is suitable to evaluate the proposed method on these two datasets.

Implementation Details We perform the proposed method based on DeepLab-v2 [Chen *et al.*, 2016a], which is the most popular framework used as the basis of many state-of-the-art scene parsing approaches. The feature learning subnet of DeepLab-v2 is transferred from the ResNet-101 [He *et al.*, 2016]. We add the master SRR module after the feature maps learned from the ResNet-101, *i.e.* the res5c layer. The two auxiliary SRR modules are applied after the 3rd and 4th blocks, *i.e.* the res3b3 and res4b22 layers. In the bottle-neck structure, channel dimension is reduced to k=512. Post-processing of dense CRF is not applied.

During training, the entire framework is trained end-to-end by the objective function in Equation (10), where we set the boundary radius $r = 5$, loss weights λ_1=λ_2=1, λ_3=0.5, γ_1=2, γ_2=1 empirically. We adopt the standard stochastic gradient descent (SGD) with the mini-batch of 4 samples. The learning rate is maintained at 0.0005 for 60 epochs. We randomly crop samples of 500×500 from images, and apply horizontal flip and random resizing between 0.5 and 1.5. We evaluate the inference speed with one image per batch averaged on all images in the validation set. Our experiments are implemented based on the MXNet platform and performed on NVIDIA Tesla K40 GPUs.

4.2 Results on Cityscapes Dataset

Ablation Study of Performance We evaluate the accuracy and inference time of the proposed HRFR method for step-by-step results on Cityscapes validation set with DeepLab-v2 framework. The down-sampling factor is set to 1/2. We also provide the results of two DeepLab-v2 models respectively trained on original-sized input (*i.e.*the teacher network) and 1/2 down-sampled input for comparison. As concluded from Table 1, we can infer that: (1) SRR with/without bottle-neck: comparing the run 1/2-HRFR(Neck) with 1/2-HRFR (Non-neck), SRR with bottle-neck saves about 91 ms (relative 11.7%), with only 0.1% reduction of accuracy which can be ignored. It verifies that SRR with bottle-neck structure can obtain faster inference speed with only very slight degradation of accuracy. (2) Auxiliary SRR and boundary region re-weighting: From the run 1/2-HRFR(Neck+Aux) vs. 1/2-HRFR(Neck), and 1/2-HRFR(Neck+Bound) vs. 1/2-HRFR (Neck), we can see that applying the auxiliary SRR modules and the boundary region re-weighting can bring 0.93% and 1.04% improvement of accuracy respectively, while employing them together can gain 1.75% improvement (1/2-HRFR (Neck+Aux+Bound) vs. 1/2-HRFR(Neck)). In summary, combination of the auxiliary SRR and boundary region re-weighting can considerably improve parsing accuracy. Additionally, the introductions of both do not affect the inference speed due to both are only adopted during training. (3) Overall HRFR verification: The proposed HRFR method (1/2-HRFR(Neck+Aux+Bound) with 1/2 down-sampled input can finally achieve 3.0× speedup compared with the teacher network with original-sized input (1-DeepLab-v2), with only 0.72% degradation of accuracy. Furthermore, compared with 1/2-DeepLab-v2 with 1/2 down-sampled input, our HRFR can gain obvious accuracy improvements of 3.79% with the same down-sampled input at merely cost of 125.7 ms mainly for SRR. Qualitative results are shown in Figure 4, which represent that boundaries in the result of the proposed HRFR method are more precise than that of 1/2-DeepLab-v2 with down-sampled input, and are comparable with that of the teacher network with original-sized input (1-DeepLab-v2). In summary, the above ablation experimental results show the effectiveness and advantages of our proposed HRFR method.

Experiments of Down-sampling Factors Table 2 demonstrates the accuracy and inference time of the proposed HRFR method by different down-sampling factors, which are also evaluated on Cityscapes validation set based on DeepLab-v2 framework. DeepLab-v2 with original-sized input provides the higher bound of accuracy (74.33%), while the 1/2, 1/4 and 1/8 down-sampled inputs for DeepLab-v2 give the higher bounds of inference speed to the corresponding down-sampling factors. For DeepLab-v2, as the down-sampling

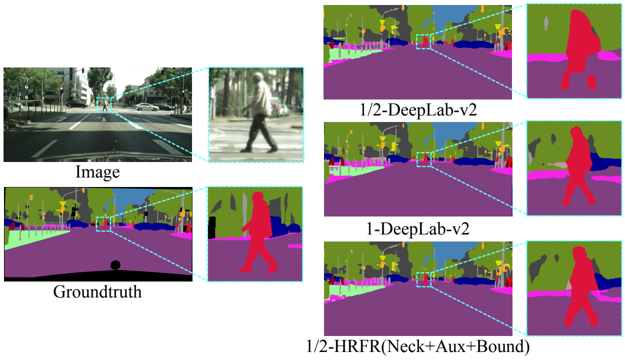

Figure 4: Qualitative results of the proposed HRFR method. Results of local regions are provided in the boxes.

Method	Down	Mean IoU(%)	Time (ms)
DeepLab-v2[Chen *et al.*, 2016a]	1/2	71.4	652.9
FRRN-B[Pohlen *et al.*, 2017]	1/2	71.8	696.8
Dilation-10[Yu and Koltun, 2016]	1	67.1	3549.5
Resnet-38[Wu *et al.*, 2016b]	1	78.4	3098.9
PSPNet[Zhao *et al.*, 2017]	1	78.4	2647.4
SegNet[Badrinarayanan *et al.*, 2017]	1/4	57.0	89.2
ENet[Paszke *et al.*, 2016]	1/2	58.3	19.3
HRFR (based on DeepLab-v2)	1/2	74.4	778.6

Table 3: Comparison with other scene parsing methods on Cityscapes test set, both accuracy and inference time are provided. Down: down-sample ratio of input images.

factor decreases (from 1/2 to 1/8), the inference speed increases exponentially (with inference time from 652.9 ms to 73.2 ms) but the accuracy decreases quickly (from 69.82% to 45.02%). By contrast, for our proposed HRFR method, accuracy decreases much slower (from 73.6% to 59.73%) as the down-sampling factor decreases, outperforming that of DeepLab-v2 by the same down-sampling factors. Based on the inference time of DeepLab-v2 with original-sized input, our proposed method can gain higher speedup rate (from 3× to 9.4×) by smaller down-sampling factors (from 1/2 to 1/8). However, as the down-sampling factor gets smaller, more layers are needed in SRR module to enlarge the resolution of feature maps. Thus the additional time for SRR module increases, which means the down-sampling factor should not be too small. In practice, the down-sampling factor of our proposed method can be determined by considering the trade-off between parsing accuracy and inference speed. In addition, the proposed method is orthogonal to the most of the existing network accelerating methods and can gain further acceleration through combining these techniques.

Comparison with previous methods We test some popular scene parsing methods on NVIDIA Tesla K40 GPUs to compare with our HRFR method on Cityscapes test set. As shown in Table 3, some methods with original-sized input and large networks can obtain a higher accuracy, but has a low inference speed, while the methods with down-sampled input and small networks have a higher speed but sacrifice the parsing accuracy. The proposed HRFR framework can make a trade-off between accuracy and speed. We believe our framework can be applied in these popular methods to speed up the methods with original-sized input or to improve the accuracy of methods with down-sampled input.

Method	CA(%)	PA(%)	Time(ms)
1-DeepLab-v2	79.69	92.30	989.7
1/2-DeepLab-v2	75.84	91.32	271.1
1/4-DeepLab-v2	66.31	87.06	81.2
1/8-DeepLab-v2	48.68	79.53	28.4
1/2-HRFR(Non-neck)	76.88	91.57	368.6
1/2-HRFR(Neck)	76.86	91.57	324.5
1/2-HRFR(Neck+Aux)	77.79	91.81	324.5
1/2-HRFR(Neck+Bound)	77.94	91.81	324.5
1/2-HRFR(Neck+Aux+Bound)	78.83	92.08	324.5
1/4-HRFR(Neck+Aux+Bound)	75.32	91.21	145.3
1/8-HRFR(Neck+Aux+Bound)	64.47	85.10	98.4

Table 4: Accuracy and inference time of the proposed method, evaluated on CamVid validation set with DeepLab-v2 framework. CA: average per-pixel accuracy over all categories. PA: overall pixel-wise accuracy. Other abbreviations are the same with Table 1.

4.3 Results on CamVid Dataset

We also evaluate the proposed method on CamVid dataset with DeepLab-v2 framework and present the results in Table 4. We can get the similar conclusion with results on Cityscapes dataset. The bottle-neck structure can provide faster inference speed while maintaining the parsing accuracy, saving the inference time of about 44 ms (relative 13.6%) per image(1/2-HRFR(Neck) vs. 1/2-HRFR(Non-neck)). In the ablation results of step-by-step comparison, both auxiliary SRR modules and boundary region re-weighting can improve the parsing accuracy by 0.93% (1/2-HRFR(Neck+Aux) vs. 1/2-HRFR (Neck)) and 1.08% (1/2-HRFR(Neck+Bound) vs. 1/2-HRFR (Neck)) in terms of average category accuracy, further combining them together can achieve the accuracy improvement of 1.97% in total (1/2-HRFR(Neck+Aux+Bound) vs. 1/2-HRFR(Neck)). The proposed HRFR method can finally achieve 2.9× speedup with 1/2 down-sampled input with only 0.86% degradation of accuracy compared with the teacher network with original-sized input (1/2-HRFR(Neck+Aux +Bound) vs. 1-DeepLab-v2). More 6.8× speedup can be obtained with 1/4 down-sampled input, but has a large degradation of 4.37% in averaged category accuracy (1/4-HRFR(Neck+Aux+Bound) vs. 1-DeepLab-v2).

5 Conclusion

In this paper, we propose the HRFR framework to speed up a given urban scene parsing network. In our framework, we exploit the SRR module to up-sample feature maps learned from the student network with down-sampled input, aiming at recovering the high resolution features from the teacher network with original-sized input. Therefore, the proposed framework can achieve the fast parsing with down-sampled input while maintaining high accuracy with original large-input network. We also present the auxiliary intermediate supervision and the boundary region re-weighting to facilitate the optimization of the network. Experimental results show the effectiveness of our proposed method.

References

[Ba and Caruana, 2014] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

[Badrinarayanan *et al.*, 2017] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *TPAMI*, 2017.

[Brostow *et al.*, 2009] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009.

[Chen *et al.*, 2016a] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[Chen *et al.*, 2016b] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016.

[Cordts *et al.*, 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[Courbariaux *et al.*, 2015] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NIPS*, 2015.

[Denton *et al.*, 2014] Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.

[Ghiasi and Fowlkes, 2016] Golnaz Ghiasi and Charless C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016.

[Han *et al.*, 2016] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Iandola *et al.*, 2016] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[Jaderberg *et al.*, 2014] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *BMVC*, 2014.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[Li *et al.*, 2017] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017.

[Liang *et al.*, 2016] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph LSTM. In *ECCV*, 2016.

[Lin *et al.*, 2016] Guosheng Lin, Chunhua Shen, Ian D. Reid, and Anton van den Hengel. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.

[Liu *et al.*, 2015] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

[Mostajabi *et al.*, 2015] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[Paszke *et al.*, 2016] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[Pohlen *et al.*, 2017] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, 2017.

[Romero *et al.*, 2015] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[Shelhamer *et al.*, 2017] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2017.

[Shuai *et al.*, 2016] Bing Shuai, Zhen Zuo, Gang Wang, and Bing Wang. Dag-recurrent neural networks for scene labeling. In *CVPR*, 2016.

[Soudry *et al.*, 2014] Daniel Soudry, Itay Hubara, and Ron Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *NIPS*, 2014.

[Vemulapalli *et al.*, 2016] Raviteja Vemulapalli, Oncel Tuzel, Ming-Yu Liu, and Rama Chellapa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, 2016.

[Wu *et al.*, 2016a] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *CVPR*, 2016.

[Wu *et al.*, 2016b] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[Xie *et al.*, 2014] Wenxuan Xie, Yuxin Peng, and Jianguo Xiao. Semantic graph construction for weakly-supervised image parsing. In *AAAI*, pages 2853–2859, 2014.

[Yang *et al.*, 2015] Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alexander J. Smola, Le Song, and Ziyu Wang. Deep fried convnets. In *ICCV*, 2015.

[Yu and Koltun, 2016] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[Zhang *et al.*, 2017a] Rui Zhang, Sheng Tang, Min Lin, Jintao Li, and Shuicheng Yan. Global-residual and local-boundary refinement networks for rectifying scene parsing predictions. In *IJCAI*, 2017.

[Zhang *et al.*, 2017b] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017.

[Zhao *et al.*, 2017] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.