SCIENCE CHINA
Information Sciences

• MOOP •

December 2017, Vol. 60 123101:1–123101:6
doi: 10.1007/s11432-017-9252-5

Semantic segmentation of high-resolution images

Juhong WANG, Bin LIU & Kun XU*

Department of Computer Science and Technology, Tsinghua University, Beijing 100084*, China*

Received May 28, 2017; accepted October 13, 2017; published online November 7, 2017

Abstract Image semantic segmentation is a research topic that has emerged recently. Although existing approaches have achieved satisfactory accuracy, they are limited to handling low-resolution images owing to their large memory consumption. In this paper, we present a semantic segmentation method for high-resolution images. First, we downsample the input image to a lower resolution and then obtain a low-resolution semantic segmentation image using state-of-the-art methods. Next, we use joint bilateral upsampling to upsample the low-resolution solution and obtain a high-resolution semantic segmentation image. To modify joint bilateral upsampling to handle discrete semantic segmentation data, we propose using voting instead of interpolation in filtering computation. Compared to state-of-the-art methods, our method significantly reduces memory cost without reducing result quality.

Keywords image semantic segmentation, high-resolution images, joint bilateral upsampling

Citation Wang J H, Liu B, Xu K. Semantic segmentation of high-resolution images. Sci China Inf Sci, 2017, 60(12): 123101, doi: 10.1007/s11432-017-9252-5

1 Introduction

Image semantic segmentation is a research topic that has emerged recently. Early image segmentation approaches typically used color, gradient, and other low-level features to achieve segmentation. In these approaches, an image is divided into many smaller pieces without any semantic meaning. In contrast, semantic segmentation not only partitions an image into different segments, but also provides a semantic label for each segment. Semantic segmentation is of great importance for computer vision and artificial intelligence tasks, such as image understanding, image recognition, etc. Take street view images for example. There is a large amount of semantic information in street view images, such as roads, traffic signs, road signs, and buildings. The understanding and recognition of such information are important for various applications, including online maps and assisted/automatic driving, and image semantic segmentation plays a basic and important role in image understanding and recognition.

Researchers have developed a large number of image semantic segmentation methods [1–5] because of its importance in these various applications. Among these methods, largely because of the progress in deep neural network techniques, neural network-based methods [5] have achieved satisfactory results. However, neural network memory consumption has a linear relationship with image resolution, and is prohibitively large for large images. Therefore, although neural networks are effective at handling images

* Corresponding author (email: xukun@tsinghua.edu.cn)

with low resolutions, they are not able to handle high-resolution images, such as street view images with a resolution of 8192×4096.

To address this problem, we propose a semantic segmentation method for high-resolution images. The main technical contribution is the use of joint bilateral upsampling [6]. Given an input high-resolution image, we first downsample the input image to a lower resolution and then obtain a low-resolution semantic segmentation image using state-of-the-art methods. Next, we obtain a high-resolution semantic segmentation image through joint bilateral upsampling of the low-resolution semantic segmentation image. Because the pixel values in semantic segmentation images are discrete values, rather than continuous values, they cannot be interpolated. To address this issue, we propose using voting instead of interpolation in the joint bilateral upsampling computation.

Compared to state-of-the-art methods [1–5], our method consumes significantly less memory, while still generating high quality results. Additionally, our method is able to handle high-resolution input images that cannot be handled by existing approaches because of their prohibitively large memory consumption.

2 Related work

2.1 Image filtering

Image filtering is a classic and important topic in image processing. Given an image I with a resolution of $N \times N$ and a template T of size $m \times m$, image filtering is used to compute a convolution of image I using the template T. The template T is also known as a filtering kernel. Image filtering is a neighborhood operation, meaning the filtered result for a pixel not only depends on the color value of the pixel itself, but also on the color values of neighboring pixels.

The Gaussian filter is one of the most commonly used smoothing filters. It is used in many image processing applications, such as image denoising. Although Gaussian filters are effective at removing noise, they also inevitably smooth desirable edges. To address this issue, Tomasi and Manduchi [7] proposed an edge-aware filtering method called bilateral filtering. It is able to remove noise while preserving image edges. The formula for bilateral filtering is

$$J_p = \frac{1}{W_p} \sum_{q \in S} I_q G_s(\|p - q\|) G_r(\|I_p - I_q\|), \tag{1}$$

where p, q denote two pixels in the image, S denotes the neighborhood of pixel p, I_p (or I_q) denotes the color value of pixel p (or q), and J_p denotes the color value of pixel p after filtering. G_s is the spatial weight and G_r is the newly introduced data weight. Both are defined using Gaussian functions. W_p is the weight normalization factor.

Based on bilateral filtering, Kopf et al. [6] proposed joint bilateral upsampling. Given a high-resolution original image $\widetilde{I}$ and a corresponding low-resolution feature image R (e.g., a depth image, rendered image, normal image, etc.), this technique is able to produce a high-resolution feature image. In joint bilateral upsampling, the spatial weight is computed using the low-resolution feature image, whereas the data weight is computed using the high-resolution original image. Specifically, the formula is

$$\widetilde{R}_{\widetilde{p}} = \frac{1}{W_p} \sum_{q \in S} R_q G_s(\|p - q\|) G_r(\|\widetilde{I}_{\widetilde{p}} - \widetilde{I}_{\widetilde{q}}\|), \tag{2}$$

where $\widetilde{p}, \widetilde{q}$ denote the pixel coordinates in the high-resolution image, p, q denote the corresponding pixel coordinates in the low-resolution image, and $\widetilde{R}$ is the resulting high-resolution feature image. In the original bilateral filtering method, both the spatial weight and data weight are computed using the same image (i.e., the input image) with the same resolution. In contrast, in joint bilateral upsampling, the spatial weight and data weight are computed using different images with different resolutions.

2.2 Semantic segmentation

Image semantic segmentation refers to a process that partitions an image into multiple different regions and specifies a semantic keyword for each region. Semantic segmentation is also referred to as scene labeling or semantic annotation. A large number of methods have been proposed for this topic. In 2007, Carneiro et al. [1] proposed a supervised learning-based semantic segmentation method. In their approach, they trained a group of feature models for each semantic class. However, their results near segmentation boundaries are typically inaccurate. In 2009, Gould et al. [2] proposed a region-based method, which directly predicts semantic labels for pre-segmented image regions, rather than for pixels. They modeled the problem using a conditional random field and considered both image appearance and scene geometry. However, the quality of the results was largely affected by the quality of the pre-segmented image regions. In 2012, Ren et al. [3] presented a semantic segmentation algorithm for RGBD (color + depth) images, which achieved a labeling accuracy of 76.1% on the NYU Depth dataset. However, their method is limited to RGBD images of indoor scenes. In 2013, Farabet et al. [4] proposed a semantic segmentation method based on hierarchical features. The main idea was to train a multi-scale convolutional neural network (CNN) directly on the pixels. The method performed well on several public datasets and required approximately one second to process an image with a resolution of 320×240. In 2015, Long et al. [5] proposed an image semantic segmentation method based on a fully CNN. The core of the method was to build a fully CNN and directly perform pixel-to-pixel, end-to-end training. The main weakness of this method is its scalability. Its memory consumption is extremely large and linear to the size of the image. In 2016, Zhou et al. [8] released a dataset, named the ADE20K dataset, for image semantic segmentation. The dataset contains approximately 20000 images with 150 labeled object types. Each image was carefully segmented into regions and each region was given an object category. They also provided a benchmark for existing semantic segmentation methods on their dataset. Note that existing approaches [1–5] mainly are focused on improving segmentation accuracy, while our approach is focused on reducing memory cost. Semantic segmentation approaches are potentially useful for various applications, including foreground extraction [9], background substitution [10], matting [11], and cloth changing [12].

3 Algorithm

Semantic segmentation algorithms are able to automatically segment images into regions and identify the object types in each region, such as sky, buildings, trees, roads, pedestrians, etc. State-of-the-art methods are able to achieve high accuracy. However, they are unable to handle high-resolution images due to their prohibitively large memory consumption. In this section, we will describe how to handle high-resolution images using our method.

3.1 Algorithm pipeline

The input for our algorithm is a high-resolution image. An example of a typical input would be a street view panorama with a resolution of 8192×4096. The algorithm uses three steps. First, we downsample the high-resolution input image to a lower resolution. Next, we generate a low-resolution semantic segmentation image from the low-resolution input image using existing methods. Finally, we generate a high-resolution semantic segmentation image using our modified joint bilateral upsampling method. We will describe the details of the algorithm in the following subsections.

3.2 Image downsampling and semantic segmentation

We have tested our algorithm on street view panoramas and high resolution indoor videos. For street view panoramas with a resolution of 8192×4096, we downsample the input images to a resolution 800×400. After obtaining the low-resolution street view panoramas, we use the CNN-based method [5] to generate semantic segmentation images. Please refer to the original paper [5] for implementation details. For the indoor scene videos with a resolution of 1600×900, we downsample all the frames to a resolution of

640×360. The low-resolution semantic segmentation of each frame is then computed using a CNN [5] and the method described in [13].

3.3 Modified joint bilateral upsampling

To upsample the low-resolution semantic segmentation images, an intuitive solution is to directly use the original joint bilateral upsampling method [6], which is computed by

$$\widetilde{R}_{\widetilde{p}} = \frac{1}{W_p} \sum_{q \in S} R_q G_s(\|p - q\|) G_r(\|\widetilde{I}_{\widetilde{p}} - \widetilde{I}_{\widetilde{q}}), \tag{3}$$

where $\widetilde{R}, R$ denote the high-resolution and low-resolution semantic segmentation images, respectively, $\widetilde{I}$ denotes the high-resolution input image, $\widetilde{p}, \widetilde{q}$ denote pixel coordinates in high-resolution images, and p, q denote pixel coordinates in low-resolution images.

However, directly using (3) is problematic. In (3), the high-resolution semantic segmentation values $\widetilde{R}$ are computed by interpolating the low-resolution values R. However, semantic segmentation values cannot be interpolated. For example, assume we use $R_q = 1$ for the semantic label "sky", $R_q = 2$ for the semantic label "ground", and $R_q = 3$ for the semantic label "building". If we directly use (3) we may produce meaningless floating-point values (e.g., $\widetilde{R}_{\widetilde{p}} = 1.5$), even if we obtain an integer value, it may still be wrong (e.g., getting $\widetilde{R}_{\widetilde{p}} = 2$ (ground) by interpolating $R_q = 1$ (sky) and $R_q = 3$ (building)).

To address this problem, we have slightly modified the joint bilateral upsampling method. The core idea is to use voting instead of interpolation. Assume that we have n different semantic labels and the semantic label R_q for each pixel is restricted to be selected from the set $N = \{1, 2, \ldots, n\}$. For each pixel, we compute the joint weight for each possible semantic label:

$$w_i = \sum_{q \in S(R_q == i)} G_s(\|p - q\|) G_r(\|\widetilde{I}_{\widetilde{p}} - \widetilde{I}_{\widetilde{q}}), \tag{4}$$

where i is in the range of 1 to n, $1 \leqslant i \leqslant n$. w_i is the joint weight of semantic label i. Finally, the semantic label of pixel $\widetilde{p}$ is simply set to that with the largest joint weight:

$$\widetilde{R}_{\widetilde{p}} = \arg\max_i w_i. \tag{5}$$

4 Comparison and results

We implemented our algorithm on a PC with an Intel Xeon 3.30 GHz CPU, 16 GB RAM, and an NVIDIA Tesla K20Xm GPU with 6 GB of GPU memory.

Figure 1 presents two frames of an indoor video. Each frame has a resolution of 1600×900. For each frame, we downsample it to a resolution of 640×480 and then use our modified joint bilateral upsampling method to produce a semantic segmentation image at the original resolution (Figure 1(d)). For comparison purposes, we also provide the high-resolution semantic segmentation results (Figure 1(b)) and low-resolution semantic segmentation results (Figure 1(c)) generated by the CNN-based method [5, 13]. Note that the results of our method (Figure 1(d)) are of similar quality to the high-resolution results (Figure 1(b)) of [5, 13], but our method requires much less memory (i.e., our method requires 1.6 GB, while the CNN-based method requires 4.9 GB for the high-resolution images).

Figure 2 presents five more examples of street view panoramas, each of which has a resolution of 8192×4096. For each example, we downsample it to a resolution of 800×400. We present our high-resolution semantic segmentation results (Figure 2(c)) and the low-resolution semantic segmentation results (Figure 2(b)) generated by the CNN-based method [5]. The CNN-based method [5] cannot handle the original high-resolution panoramas because the memory requirements exceed the GPU memory limit. Note that our method uses the same amount of memory to handle high-resolution images as the CNN-based method uses to handle low-resolution images (i.e., both use 1.9 GB of memory), while our method

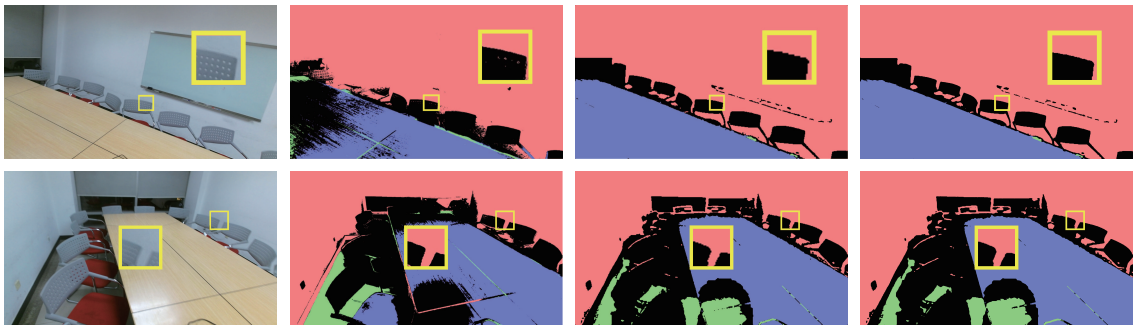

Figure 1 (Color online) Examples of an indoor video: (a) two frames of the input video; (b) high-resolution semantic segmentation results by [5, 13]; (c) low-resolution semantic segmentation results by [5, 13]; (d) our high-resolution semantic segmentation results.

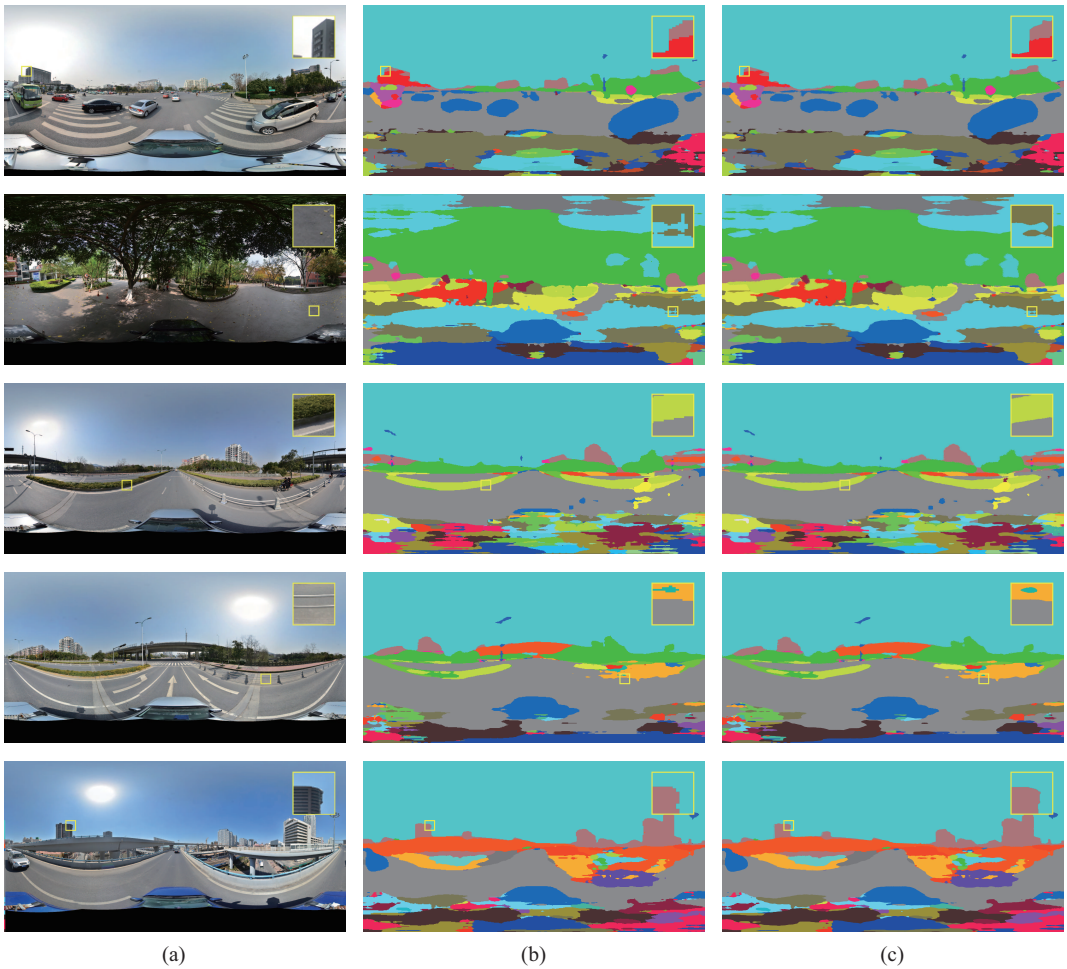

Figure 2 (Color online) Examples of street view panoramas: (a) input images; (b) low-resolution semantic segmentation results by [5]; (c) our high resolution semantic segmentation results.

achieves superior results because it can handle images with higher resolutions (see the enlarged views in Figure 2(b) and (c)).

In Table 1, we provide statistics, including the resolution of semantic segmentation result images and GPU memory consumption, for each method for all examples.

Table 1 Statistics

Example	Our method		CNN method, high-resolution		CNN method, low-resolution	
	Resolution	Memory (G)	Resolution	Memory (G)	Resolution	Memory (G)
Indoor, Figure 1	1600×900	1.6	1600×900	4.9	640×480	1.6
Panorama, Figure 2	8192×4096	1.9	N/A	N/A	800×400	1.9

5 Conclusion

We presented a semantic segmentation method for high-resolution images. First, we downsample the input image to a lower resolution and then obtain a low-resolution semantic segmentation image using state-of-the-art methods. Next, we use a modified joint bilateral upsampling method to generate a high-resolution semantic segmentation result. Our method significantly reduces memory consumption and performs well on high-resolution images.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61521002), a research grant from the Beijing Higher Institution Engineering Research Center, and the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

Conflict of interest The authors declare that they have no conflict of interest.

Supporting information The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Carneiro G, Chan A B, Moreno P J, et al. Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans Pattern Anal Mach Intell, 2007, 29: 394–410
2 Gould S, Fulton R, Koller D. Decomposing a scene into geometric and semantically consistent regions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Kyoto, 2009. 1–8
3 Ren X, Bo L, Fox D. RGB-(D) scene labeling: features and algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, 2012. 2759–2766
4 Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell, 2013, 35: 1915–1929
5 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015. 3431–3440
6 Kopf J, Cohen M F, Lischinski D, et al. Joint bilateral upsampling. ACM Trans Graph, 2007, 26: 96
7 Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Bombay, 1998. 839–846
8 Zhou B, Zhao H, Puig X, et al. Scene parsing through ADE20K dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017
9 Li X, Liu K, Dong Y. Superpixel-based foreground extraction with fast adaptive trimaps. IEEE Trans Cybern, 2017, doi: 10.1109/TCYB.2017.2747143
10 Huang H, Fang X, Ye Y, et al. Practical automatic background substitution for live video. Comp Visual Media, 2017, 3: 273–284
11 Li X, Liu K, Dong Y, et al. Patch alignment manifold matting. IEEE Trans Neural Netw Learn Syst, 2017, doi: 10.1109/TNNLS.2017.2727140
12 Zheng Z H, Zhang H T, Zhang F L, et al. Image-based clothes changing system. Comput Vis Media, 2017, in press
13 Maerki N, Perazzi F, Wang O, et al. Bilateral space video segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016. 743–751