SEMANTIC SEGMENTATION VIA SPARSE CODING OVER HIERARCHICAL REGIONS

Wenbin Zou, Kidiyo Kpalma and Joseph Ronsin
Université Européenne de Bretagne, France
INSA, IETR, UMR CNRS 6164
FirstName.LastName@insa-rennes.fr

ABSTRACT

The purpose of this paper is segmenting objects in an image and assigning a predefined semantic label to each object. There are two areas of novelty in this paper. On one hand, hierarchical regions are used to guide semantic segmentation instead of using single-level regions or multi-scale regions generated by multiple segmentations. On the other hand, sparse coding is introduced as high level description of the regions, which contributes to less quantization error than traditional bag-of-visual-words method. Experiments on the challenging Microsoft Research Cambridge dataset (MSRC 21) show that our algorithm achieves state-of-the-art performance.

Index Terms— Semantic segmentation, sparse coding, hierarchical regions, image understanding

1. INTRODUCTION

Image segmentation has been studied for several decades. The traditional region-based segmentation techniques such as graph cuts [1][2], normalized cuts [3], and mean-shift [4] belong to bottom-up approach, where homogeneous pixels are grouped together based on low-level features, e.g. texture, color and boundary continuity. However, this approach is still far from to satisfy accurately segmenting objects. Indeed, in most cases objects are over-segmented into several regions. Recently, there has been growing interests in semantic image segmentation, which combines the segmentation together with object recognition and leads to partitioning an image into its constituent objects and assigning a semantic label to each object.

One of the popular approaches for semantic segmentation is using the low-level segmentation as the guidance of high-level description. Yang et al. [5] computed bag-of-keypoints on over-segmented mean-shift regions. Csurka and Perronnin [9] also used mean-shift segmentation and deployed Fisher description over each region. As over-segmentation might result in noisy partition, some authors proposed creating regions with multiple segmentations. Russell et al. [6] performed normalized cuts algorithm 12 times on an image to generate 96 overlapping regions, and applied Probabilistic Latent Semantic Analysis (pLSA) to detect the objects in a set of images. Pantofaru et al. [11] even did more, using three segmentation algorithms [2][3][4] (up to 18 segmentations) to generate a set of overlapping regions and bag-of-visual-words (BOV) model associated as high-level representation. Indeed, the more segmentation is used, generally, the more chance one has to capture objects in an image. But it also increases computational complexity to $O(n)$, where n is the number of segmentations.

In this paper, we investigate to make use of hierarchical segmentation. Compared to multiple segmentations, it does not increase the computational complexity while providing hierarchical regions. Another contribution of this paper is the introduction of sparse coding as the high-level representation. While it has been shown to lead to high accuracy of image classification [21], the sparse coding has not been applied to semantic image segmentation. We demonstrate that, even without using any random field models which are widely used in recent approaches to incorporate multi-cues, our algorithm obtains state-of-the-art results on the standard dataset of semantic segmentation.

This rest of the sections are organized as follows. Section 2 introduces the proposed algorithm in detail which includes creating hierarchical regions, local feature extraction, sparse-based high-level description, region scoring and labeling. Section 3 presents experimental results and compares ours with those of recent approaches. Finally, section 4 concludes the paper.

2. PROPOSED ALGORITHM

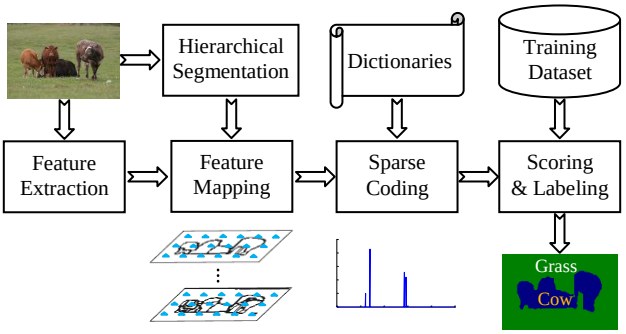


Figure 1. **Framework of semantic segmentation**

As showed in figure 1, a segmentation algorithm is used to generate hierarchical regions. In the meanwhile, local features are extracted from the input image, and mapped to

each region. Then the sparse-based high-level descriptors are computed for each region and subsequently scored by discriminative classifiers. Finally, object class labels are assigned to each pixel by observing the classification scores and spatial correlation.

2.1. Generating hierarchical regions

To generate hierarchical regions, we prefer the algorithm proposed in [15] because it generally preserves global contours of objects leading to natural constraints for feature extraction. The output of this segmentation is a valued Ultrametric Contour Map (UCM), where contour values reflect contrast between neighboring regions. Hierarchical regions can be created by thresholding the UCM with different thresholds. The key problem of thresholding is how to define the thresholds. Consider the fact that over-segmentation might lead to noisy labeling and under-segmentation might result in two or more objects merging into the same region, the thresholds should neither be set too small nor too large. In addition, it is inadvisable to fix arbitrarily minimum and maximum thresholds, because the contour values in UCM strongly depend on luminance and contrast of the image. Therefore, we design a self-adapting approach to define the range of thresholding, where minimum and maximum thresholds are proportional to the maximum contour values

$$Thr_{min} = \alpha * \max(UCM) \quad (1)$$
$$Thr_{max} = \beta * \max(UCM) \quad (2)$$

here $\alpha, \beta \in (0,1)$ are predefined parameters. In our experiments, α and β are set to 0.25 and 0.8 respectively. Contour values in this range are taken as the thresholds to create hierarchical regions. Typically we obtain 5 to 20 thresholds per image. Even such strategy cannot totally avoid the problem mentioned above; we will consider this aspect during the region labeling stage. Unlike multi-segmentation approaches that increase computational cost multi-folds, the thresholding process adds hardly any computational burden.

2.2. Feature extraction

This section briefly introduces two kinds of local features used in experiments of section 3.

The first one is Scale-Invariant Feature Transform (SIFT) [19]. SIFT descriptors are extracted on a regular grid with a step-size of 6 pixels. For each grid, the SIFT descriptors are computed respectively at four scales (4, 8, 12, 16 pixel radii). And these descriptors are computed for each RGB component. One SIFT descriptor is represented with a 3x128 dimensional vector.

The second is self-similarity feature (SSIM) [20]. SSIM descriptors are extracted from a regular gird with step-size of 4 pixels. The SSIM descriptor is generated by computing correlation map of 5×5 pixels patch in a surrounding 20×20 pixels patch, and then quantizing it into 40 bins (10 angles, 4 radial intervals). Hence one SSIM descriptor is a 40 dimensional vector.

Both SIFT and SSIM features are extracted in a dense approach instead of sparse approach which only computes descriptors on keypoints. This is because keypoint detectors generally have difficulties to detect keypoints in uniform regions, such as sky, calm water and road, and lead to unassignment on these areas. The local feature vectors are computed over the entire image and then projected to each region of the image.

2.3. High-level description

To transform local features into high-level description, traditional approach of BOV model is based on visual dictionary: each local feature vector is represented with the nearest basic vector of the dictionary. However, this approach results in quantization error because only a single basic vector is used to represent a local feature vector. To solve this problem, we introduce sparse-based high-level description.

Given a set of local feature vectors $X = [x_1, x_2 \dots, x_N]$ in $\mathbb{R}^{M\times N}$, our purpose is to construct a dictionary $D = [d_1, d_2 \dots, d_K]$ in $\mathbb{R}^{M\times K}$, where each column represents a basic vector, and to describe each local feature vector approximately as a weighted linear combination of a few basic vectors

$$x_n \cong Da_n \quad (3)$$
$$s.t.\ a_n \succeq 0,\ \forall n = 1, \dots, N$$

where a_n in $\mathbb{R}^{K\times 1}$ is coefficient vector, in which most entries are zeros; $a_n \succeq 0$ denotes all elements in a_n are non-negative. Solving this problem is equivalent to optimize the cost function

$$f(D, A) = \min_{D,A} \sum_{n=1}^{N} \|x_n - Da_n\|_2^2 \quad (4)$$
$$s.t.\ a_n \succeq 0,\ \ \forall n = 1, \dots, N$$

where $A = [a_1, a_2 \dots, a_N]$ in $\mathbb{R}^{K\times N}$; $\|\cdot\|_2$ is the ℓ_2 norm of vector. To do this we apply positive constrained sparse coding [16] to (4)

$$\min_{D,A} \sum_{n=1}^{N} \|x_n - Da_n\|_2^2 + \lambda\|a_n\|_1 \quad (5)$$
$$s.t.\ \|d_k\|_2 \le 1, \forall k = 1, \dots, K,\ a_n \succeq 0, \forall n = 1, \dots, N$$

where λ is a regularization parameter. ℓ_1 regularization produces sparse coefficients for a_n [17]. Constraining ℓ_2 norm of vector d_k less or equal to one is to prevent D from arbitrarily large values which would due to arbitrarily small values of A. The dictionary D is obtained by minimizing (5) with respect to D and A (i.e. alternatively minimizing over one while keeping the other one fixed). Once dictionary D is constructed, sparse coefficient vector can be computed by minimizing (5) only with respect to A. Accordingly, each local feature vector can be approximated by multiplying the

dictionary *D* and a sparse coefficient vector. In other words, sparse coding represents one local feature vector with a linear combination of a few basic vectors. We have compared reconstruction performance of sparse coding and BOV methods. The former decreases the Mean Squared Error (MSE) from 6.4 to 2.6 corresponding to 59% reduction in case of reconstructing SIFT feature with a dictionary containing 2000 basic vectors.

As two kinds of local features (SIFT and SSIM) are used in our algorithm. Similar to BOV, a subset of local feature vectors is randomly chosen to train SIFT and SSIM sparse dictionaries respectively with 2000 and 800 basic vectors (these values are determined experimentally). Then the dictionaries are used to compute sparse vectors of regions.

2.4. Region scoring

We now classify sparse coded regions to relevant object classes. Theoretically, any discriminative classifier may be performed on this task. In this study, we prefer Support Vector Machine (SVM) with Multiple Kernel Learning (MKL) [18], as it is easy to train classifiers incorporating several kinds of features even that these features are mapped by different kernels.

For classification, we firstly compute normalized histogram of sparse vectors for each region

$$h_i = \frac{1}{J_i}\sum_{j=1}^{J_i} a_j \qquad (6)$$

where $a_j, j = 1, \dots, J_i$, denotes sparse vectors in region R_i, and each sparse vector is normalized to sum to unity. By using (6), we can compute the histogram of SIFT sparse vectors denoted as h_i^t, and that of SSIM sparse vectors denoted as h_i^m. $h_i^c = \{h_i^t, h_i^m\}$ is defined as the combination of feature histograms. So the classification function of a SVM in kernel formulation is expressed as:

$$SVM(h^c) = \sum_{i=1}^{I} y_i\, \alpha_i K(h^c, h_i^c) + b \qquad (7)$$

where $h^c$is feature histogram of a test region; $h_i^c, i = 1, \dots, I$, are feature histograms of *I* training regions; $y_i \in \{+1, -1\}$ indicate their class label; and K is positive definite kernel, which is calculated as a linear combination of feature histogram kernels

$$K(h^c, h_i^c) = d^t K(h^t, h_i^t) + d^m K(h^m, h_i^m) \qquad (8)$$

where d^t and d^m denote nonnegative kernels weights. Many kernels can be applied for the histogram-based classification, such as intersection kernel, Chi2 kernel and RBF kernel. In our experiments, Chi2 kernel is used for both the histograms of SIFT and SSIM. MKL learns the weights d^t, d^m and parameters α_i, b for each class. By using (7) a test region can obtain a SVM score, indicating the likelihood of object class, from each classifier.

2.5. Region labeling

The most direct approach for labeling scored regions of a test image is to assign these regions with the most likely class labels. However it cannot be directly applied to our algorithm, because the hierarchical regions are overlaid or crossed with each other; in addition, as mentioned in subsection 2.1, those regions generated by coarse thresholding might merge several objects. Our solution is to combine SVM scores with sizes of regions.

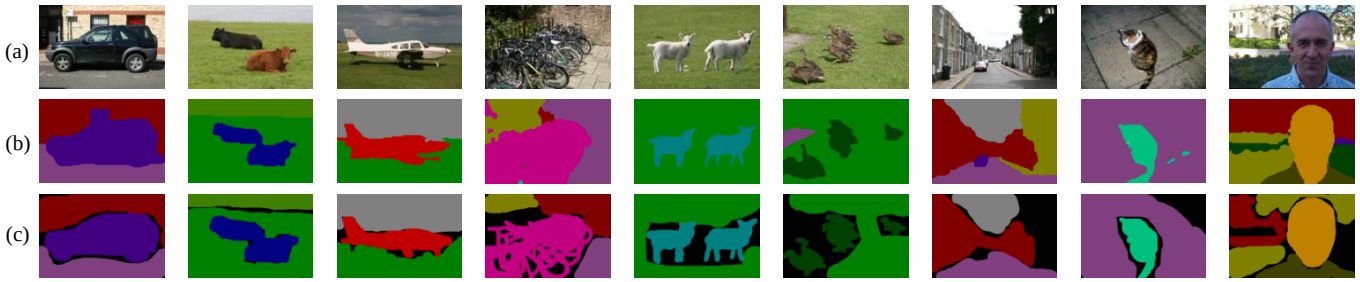
The labeling process mainly consists of three steps. Firstly, the most likely object classes that have the maximum SVM scores are used to pre-label each region. Secondly, these regions are sorted by their increasing SVM scores. Finally, the regions are gradually merged to form a complete labeled image by observing their sizes and SVM scores. Concretely, when a candidate region R_j or its part locates at the same position as labeled region R_i, only its score great enough and it is not much larger than R_i, it can overwrite the region R_i. This strategy avoids labeling small objects as their surrounding environment or neighboring large objects.

3. EXPERIMENTS

In this section, we evaluate our method on the standard dataset of semantic segmentation: MSRC 21[8]. This dataset contains 591 color images of 21 object classes. Each image has ground-truth segmentation that uses different colors to label each pixel with one of 21 object classes or void (in black). We use the same splitting protocol as in [9][10]: 276 images for training and the rest 315 images for testing. Segmentation performance is measured by both average accuracy (defined as average label accuracy across all object classes) and global accuracy (defined as percentage of all test image pixels assigned to the correct class label). Some examples of segmentation are presented in figure 2. As showed in figure 2, the average accuracy of the proposed algorithm is 73%, slightly lower than 75% reported in [14]; however, the global accuracy we obtain is 82% which is about 4% improvement over the state-of-the-art.

4. CONCLUSION

This paper presents a novel semantic segmentation algorithm. Hierarchical regions are used to guide features extraction. Sparse coding is introduced as high-level representation for semantic segmentation which contributes to less quantization error than traditional BOV model. Experimental results show that the proposed approach obtains state-of-the-art performance on the MSRC 21 dataset.



	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bike	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	**Average**	**Global**
Shotton et al. [7]	49	88	79	**97**	**97**	78	82	54	87	74	72	74	36	24	93	51	78	75	35	**66**	18	67	72
Csurka et Perronnin [9]	**84**	95	81	67	78	89	72	77	87	71	86	66	59	28	85	19	68	59	47	35	9	65	77
Lim et al.[12]	30	71	69	68	64	84	**88**	58	77	82	91	90	**82**	34	93	74	31	56	54	54	**49**	67	-
Jiang et Tu [13]	53	**97**	83	70	71	**98**	75	64	74	64	88	67	46	32	92	61	**89**	59	66	64	13	68	78
Gonfaus et al. [14]	60	78	77	91	68	88	87	76	73	77	**93**	**97**	73	**57**	**95**	**81**	76	**81**	46	56	46	**75**	77
Proposed method	74	90	**84**	72	83	84	76	**83**	**90**	**89**	80	94	76	43	88	46	72	63	**73**	53	24	73	**82**

Figure 2. **Segmentation results from MSRC 21 dataset. Above:** (a) original images; (b) segmented results; (c) ground-truth segmentation. **Below**: segmentation accuracies (percent) over the whole dataset. The highest accuracies are remarked in bold.

5. REFERENCES

[1] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in Proc. *ICCV*, July 2001.

[2] P. Felzenszwalb, D. Huttenlocher, "Efficient graph-based image segmentation," *Int'l J. of Computer Vision*,Vol. 59, No. 2, Sep. 2004.

[3] J. Shi and J. Malik "Normalized cuts and image segmentation," in Proc. *CVPR*, 1997.

[4] D. Comaniciu, P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," in Proc. *PAMI*, pp. 603-619, May, 2002.

[5] L. Yang, P. Meer and D. J. Foran, "Multiple class segmentation using a unified framework over mean-shift patches," in Proc. *CVPR*, June 2007.

[6] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in Proc. *CVPR*, 2006.

[7] J. Shotton, M. Johnson, R. Cipolla, "Semantic Texton Forests for Image Categorization and Segmentation," in Proc. *CVPR*, 2008.

[8] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: joint appearance, shape and context modeling for multi-class object recognition and segmentation" in Proc. *ECCV* 2006.

[9] G. Csurka, F. Perronnin, "An efficient approach to semantic segmentation," *Int'l J. of Computer Vision* pp. 1–15 (2010)

[10] J. Verbeek and B. Triggs, "Region classification with Markov field aspect models," in Proc. *CVPR*, June 2007.

[11] C. Pantofaru, C. Schmid and M. Hebert, "Object Recognition by Integrating Multiple Image Segmentations," in Proc. *ECCV*, 2008.

[12] J.J. Lim, P. Arbelaez,C. Gu, and J. Malik, "Context by region ancestry," in Proc. *ICCV*, 2009.

[13] J. Jiang, Z. Tu, "Efficient scale space auto-context for image segmentation and labeling" in Proc. *CVPR*, 2009

[14] J.M. Gonfaus, X. Boix, J. van de Weijer, A.D. Bagdanov, J. Serrat, and J. Gonzàlez, "Harmony potentials for joint classification and segmentation," in Proc. *CVPR*, 2010.

[15] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour Detection and Hierarchical Image Segmentation," in Proc. *PAMI*, Vol. 33, No. 5, pp. 898-916, May 2011.

[16] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online learning for matrix factorization and sparse coding," in *Journal of Machine Learning Research* 11(2010) 19-60.

[17] A. Y. Ng. "Feature selection, L1 vs. L2 regularization, and rotational invariance," in Proc. *ICML*, 2004.

[18] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Proc. *ICCV*, Brazil, October 2007.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in Proc. *ICCV*, Corfu, Greece (September 1999), pp. 1150-1157.

[20] E. Shechtman and M. Irani, "Matching Local Self-Similarities across Images and Videos," in Proc. *CVPR*, 2007.

[21] J. Yang, J. Wang, and T. Huang, "Learning the sparse representation for image classification," in Proc. *ICME*, 2011.