Integrating Low-level and Semantic Features for Object Consistent Segmentation

Hao Fu
School of Computer Science
University of Nottingham
Nottingham, UK
hxf@cs.nott.ac.uk

Guoping Qiu
School of Computer Science
University of Nottingham
Nottingham, UK
qiu@cs.nott.ac.uk

***Abstract*—The aim of semantic segmentation is to assign each pixel a semantic label. Numerous methods for semantic segmentation have been proposed in recent years and most of them chose pixel or superpixel as the processing primitives. However, as the information contained in a pixel or a superpixel is not discriminative enough, the outputs of these algorithms are usually not object consistent. To tackle this problem, we introduce the concept of object-like regions as a new and higher level processing primitive. We first experimentally showed that using object-like regions as processing primitives can boost semantic segmentation accuracy, and then proposed a novel method to produce object-like regions by integrating state-of-art low-level segmentation algorithms with typical semantic segmentation algorithms through a novel semantic feature feedback mechanism. We present experimental results on the publicly available image understanding database MSRC21 and show that the new method can achieve state of the art semantic segmentation results with far fewer processing primitives.**

***Keywords*-semantic segmentation; object-like regions; feedback mechanism**

I. Introduction

Holistic image understanding is always the Holy Grail in computer vision. To achieve this goal, a natural method is to adopt a segment-then-recognize strategy, i.e. first segment an image into different objects, then recognize each object one by one. However, it is generally believed that segmentation is an ill posed problem. Taking the image in Fig.1 as an example, the head of the sheep (black color) and the body of the sheep (white color) are totally different from each other. But still they belong to one object. Therefore, we could not expect our segmentation module which are purely based on low-level features can produce semantic consistent regions.

To circumvent this problem, recent methods tend to bypass the segmentation module. On one hand, sliding window based object recognition paradigm has achieved remarkable success in some specific areas, like face detection [1]; on the other hand, pixel or superpixel based semantic segmentation methods [2], [3], [4] has become the mainstream in holistic image understanding. Are we gradually losing faith in image segmentation? The answer is of course not. In [5], the authors experimentally confirmed that segment based recognition can achieve a higher accuracy than sliding window based recognition; although one segmentation is always prone to make mistakes, [6] used multiple segmentations.

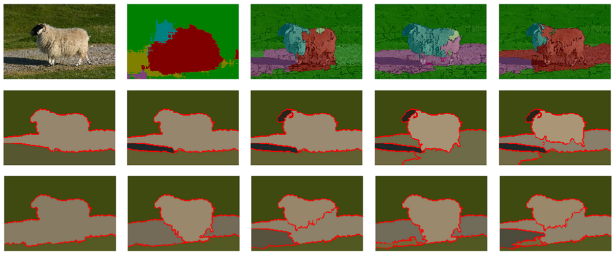

Figure 1. Top row: (left to right) original image, pixel based semantic labeling [4] without CRF [7] post processing, three groups of superpixel based methods [3] by varying the parameters of mean shift; Middle row: spectral segmentation [8] by only considering color features to produce K (K=3,4,...7) clustering; Bottom row: our full model by combining color features and semantic features to produce K clustering. It is seen that in all cases our method consider the head and the body of the sheep to be one object.

Although the use of multiple segmentations increases our chance of finding object-like regions, it's still a low-level segmentation. A better model can be expected if we could incorporate top down information into the low-level segmentation module. We named this kind of segmentation as supervised segmentation. Still taking the image in Fig.1 as an example. How can the segmentation algorithm consider the head of the sheep and the body of the sheep to be one object? The only way is to incorporate supervised information into the segmentation module. As in our training phase, probably we have already seen a sheep with black head and white body, thus it is possible to utilize these training images to guide our segmentation algorithm. Therefore, the key problem is how to train our segmentation module to achieve this.

In this paper, we proposed a method that treats the output of typical semantic segmentation algorithm as top down information to guide the bottom up unsupervised segmentation. More precisely, we treat the output of typical semantic segmentation algorithm as semantic features. These semantic features are combined with low-level features, which together determines the similarity between neighboring nodes. In this case, although the head of the sheep and the body of the sheep differs a lot in their low-level feature, they may

978-0-7695-4541-7/11 $26.00 © 2011 IEEE
DOI 10.1109/ICIG.2011.47

share similar semantic features, thus making it possible to consider these two parts belonging to one object.

In summary, we made two contributions in this paper. Firstly, we emphasized the importance of choosing object-like regions as processing primitives in image understanding. Secondly, we propose to combine state-of-art low-level segmentation with typical semantic segmentation algorithm to produce those object-like regions.

II. Related work

The problem we are interested here is holistic image understanding, which means given an image, our task is to assign a semantic label to every pixel of the image. There exists a large literature in this domain. A common aspect shared by most of the existing algorithms is that they usually choose pixel or superpixel [2], [10], [4] as their processing primitives. However, one direct consequence of taking such strategy is that the output of the algorithm is usually not object consistent (as shown in Fig.1), and each object is just a group of pixels or superpixels with the same labels.

One notable exception beyond this theme is the work in [11], where the authors directly chose the region as their processing primitives, and they achieved the best segmentation accuracy on the recent PASCAL challenge [12]. Comparing to choosing pixel or superpixel as processing primitives, choosing region enjoys many benefits: the information contained in a region is much more comprehensive than contained in a superpixel, and the shape of the region is beneficial for recognizing some shape dominate objects.

Although the strategy of choosing region as primitives is appealing, it is very difficult to segment images to semantic consistent regions in practice. [11] circumvent this problem by generating multiple figure-ground hypothesis. However, the segmentation module they adopted is still low-level cue based. Although they performed a supervised ranking after the hypothesis generation, the errors occurred in the low-level segmentation module will not be remedied. **After all, if we want to achieve a semantic consistent segmentation, we need to incorporate information beyond the image itself.** While the authors in [13] used a shape prior to guide the segmentation, classcut [14] or co-segmentation [15] utilize information from another images. Another interesting work in [16] retrieves similar images from the internet, and these retrieved images are treated as additional information to guide the segmentation.

Different from all those previous works, in this paper we aim to obtain the additional information from the training data. In a typical low-level spectral segmentation algorithm, the core module is to define the affinity weights between neighboring nodes. In order to make the low-level segmentation module produce semantic consistent regions, we had to incorporate high-level knowledge in defining these weights. As we aim to obtain these high-level knowledge from the training data, we find typical semantic segmentation algorithms exactly meets our needs: their posterior outputs can just define our semantic likeliness between neighboring nodes.

As in essence, the semantic segmentation algorithm is still based on low-level cues, our algorithm can be considered by introducing a feedback from the classifier output to the input level. We believe this idea is an important contribution of our work. As we have also noticed similar ideas being adopted in some other scenarios, we believe it could be an important philosophy could be generally adopted in many scenarios.

In [17], the outputs of various object detectors are treated as new mid-level features; in [18], the authors learned a bank of weak classifiers from the web-retrieved images, the output of these weak learners are also treated as mid-level features. In this theme, the output of semantic segmentation algorithm used in our low-level segmentation module can also be considered as mid-level features. In [19], the authors first learned a classifier on the local patch, then the output of the classifier is considered as the context information of the target patch. By concatenating them together, the author retrained the system. A similar strategy is also adopted in [20], where the authors use the geometric context detector to detect the layout of similar images, then the average of those outputs are considered as a prior, together with the original features, the author retrained the system. From those previous works, we came to a conclusion that the output of classifier contains useful information. They can enhance the performance of the original system through a carefully design. This also justifies the applicability of the framework we proposed in the next section.

III. Combining low-level and semantic segmentation for object proposal

A. The advantage of choosing object-like regions as processing primitives

Before we introduce our newly proposed framework, we would like to first emphasize the importance and advantage of choosing object-like regions as processing primitives. We choose MSRC21 as our test bed. MSRC21 contains 591 images from 21 semantic classes. Following [4], the dataset is divided into 276 images for training, 59 images for validation, and the remaining 256 images for testing. For the object-like regions, we directly use the clean ground truth segmentations[1] on this dataset. These ground truth segmentations are directly fed into our object-like regions labeling module to be described in section III-F. We obtain an overall global pixel accuracy of 90.5%, outperforming any state-of-art methods [2], [3] on this dataset. This experiment clearly shows the advantage of choosing the object-like regions as our processing primitives. However, the challenge is how to automatically generate such object-like regions. We present such a method in the following sub sections.

[1] http://www.cs.cmu.edu/ tmalisie/projects/bmvc07/

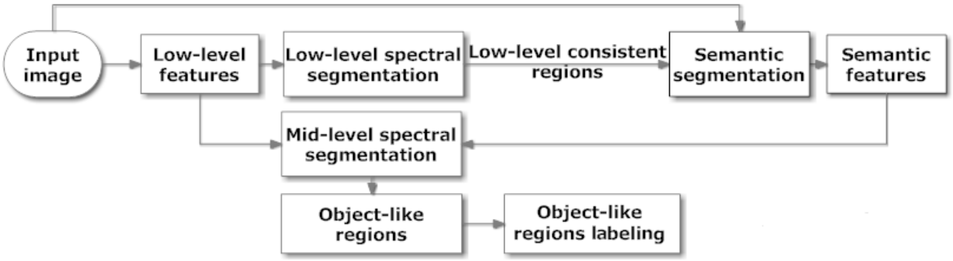
Input image
Low-level features
Low-level spectral segmentation
Low-level consistent regions
Semantic segmentation
Semantic features
Mid-level spectral segmentation
Object-like regions
Object-like regions labeling

Figure 2. The flow chart of the proposed framework

B. The proposed framework

Given an input image, several kinds of low-level features can be extracted, and these features are fed into a low-level spectral segmentation module. These low-level segmentation algorithms can produce low-level consistent regions, which we call them superpixel here. A typical semantic segmentation algorithm can either choose these superpixel or directly choose the image pixel as the processing primitives, and their outputs can be considered as semantic features.

Those semantic features are then combined with the low-level features and fed back to the low-level segmentation algorithm again. This time, as the features contain semantic feature, we call the segmentation module mid-level segmentation. Ideally, they will segment images into regions that are both low-level and semantic-level consistent, i.e. they are object-like regions. Based on those object-like regions, typical classification algorithms can be used to achieve the goal of image understanding. The flow chart of this procedure is shown in Fig.2.

C. Spectral methods for low-level segmentation

For the low-level spectral segmentation module, we adopt the multi-layer graph method proposed in [8], as it produces the state-of-art results on BSDS dataset [21]. A multi-layer graph is represented by $G^* = (V^*, E^*)$, where the nodes V^* are a set of pixels and superpixels, and edges E^* exist between neighboring pixels, neighboring superpixels, and between the superpixel with all the pixels it includes. The edge weights are defined as:

$$w_{ij} = \begin{cases} exp(-\theta_g \parallel g_i - g_j \parallel) & if \ i,j \in pixels \\ exp(-\overline{\theta}_g \parallel \overline{g}_i - \bar{g}_j \parallel) & if \ i,j \in superpixels \\ const & otherwise \end{cases} \tag{1}$$

where g_i is the color value (in Lab space) of pixel i, and $\overline{g}_i$ represents the mean color of all the inner pixels contained in a superpixel i. θ_g and $\overline{\theta}_g$ are constants that control the strengths of the weight. They can be specified either manually or using cross-validation techniques. For details of this algorithm, please refer to [8].

D. Pixel or superpixel based semantic segmentation

For the semantic segmentation module, we adopt the typical two order Conditional Random Field (CRF) [7] based methods. Mathematically speaking, a CRF defines a posterior distribution of hidden random variables Y (labels), given observed image features X, in a factored form:

$$p(Y|X) = \frac{1}{Z}\exp(-\sum_{c\in C}\psi_c(Y_c, X)) \tag{2}$$

where Z is a normalizing constant, and C is the set of all cliques. When the size of the clique c is one, $\psi_c(Y_c, X)$ corresponds to the node potential, accordingly, when its size is two, $\psi_c(Y_c, X)$ corresponds to the pairwise potential.

Given a set of images and its corresponding groundtruth labels, the training procedure of a CRF aims to make the energy of the groundtruth label assignment corresponds to the minimum of the energy function. After the model is trained, for a new test image, its most probable of labeling Y^* is defined as

$$Y^* = \arg\max_{Y\in L} p(Y|X) \tag{3}$$

where L corresponds to any kinds of possible labelings.

Besides the most probable labelling Y^*, we can also get the marginal posterior distribution $p(y_i)$ of any node i. In this paper, we decide to use this marginal distribution as our semantic feature. Comparing to directly choosing the MAP assignment as the semantic feature, it clearly enjoys some benefits. For example, for two neighboring nodes, we can not only judge if they are most likely to belong to one specific semantic class, but also we could say whether they are all dissimilar with another semantic class.

In our experiments, for the pixel based semantic segmentation, we adopted the Textonboost method [4] and use their publicly available code[2]; for superpixel based semantic segmentation, we adapted the code from the STAIR Vision Library[3], which uses piecewise training method [22] to train the CRF and uses max-product propagation [23] for inference.

[2] http://jamie.shotton.org/work/code.html

[3] http://robotics.stanford.edu/ sgould/svl/

E. Mid-level segmentation with semantic feature feedback

We propose to introduce the semantic features generated by CRF into the above mentioned low-level spectral segmentation module, to enable the spectral segmentation algorithm not only produce low-level consistent regions, but also semantically consistent regions.

One intuitive way of achieving this goal is to redefine the edge weight function of equation (1). Note that in equation (1), the weight between neighboring nodes only depends on their low-level features. By introducing our semantic segmentation module, we can also obtain the semantic feature of each node. Thus, we redefine the weight between neighboring nodes as a combination of their low-level feature similarity and their semantic feature similarity:

$$w_{ij} = \begin{cases} \exp(-\theta_g \parallel g_i - g_j \parallel) + \exp(-\theta_s \parallel s_i - s_j \parallel) \\ \qquad\qquad\qquad\qquad if \;\; i,j \in pixels \\ \exp(-\bar{\theta}_g \parallel \bar{g}_i - \bar{g}_j \parallel) + \exp(-\bar{\theta}_s \parallel \bar{s}_i - \bar{s}_j \parallel) \\ \qquad\qquad\qquad\qquad if \;\; i,j \in superpixels \\ const \qquad\qquad\qquad otherwise \end{cases} \tag{4}$$

where s_i corresponds to the semantic feature of node i, here it is equivalent to the marginal probability $p(y_i)$ of node i in our CRF paradigm. $\parallel \parallel$ represents the norm of the vector. We experimentally compared different kinds of norms and choose $L1$ norm which performs best in our experiments. As has been shown by others, $L1$ norm tends to produce better results because its robustness against outliers [24].

F. Object-like primitives labeling

After producing object-like regions using the spectral segmentation algorithm with semantic feature feedback, we now extract features of those object-like regions and build a classifier to label them. As the object-like regions are usually large and contain many pixels, we believe that their histogram features are more robust and discriminative. Similar to [25], several kinds of histogram features, including Texton Histogram, Color Histogram and pHOG Histogram are extracted from each region. All these histograms are generated by aggregating the assignments where all the features are quantized according to a pre-trained codebook. We use χ^2 kernel to measure the similarities among those histograms. A typical Multiple Kernel Learning approach[4] is then adopted to learn the weights of these kernels to achieve an optimal classification performance of the final classifier.

IV. Experiments

A. Segmentation quality

To assess the quality of the regions generated by our models, we adopt the segmentation covering [26] as an accuracy measure. The covering of a segmentation S by a segmentation S' is defined as

$$C(S,S') = \frac{1}{N} \sum_{R \in S} |R| * \max_{R' \in S'} O(R,R') \tag{5}$$

where N denotes the total number of pixels in the image, $|R|$ represents the number of pixels in the region R, and O is the overlap.

For each test image, we segmented it into a fixed number (3 to 12) of regions using different segmentation algorithms. Then we compare these segments to the ground truth segmentation to compute the segmentation covering score. Here we compared Normalized Cut [27] and Full Pairwise Affinity Model in [8] with our full model. The results are shown in Fig.3. From there we can see our full model consistently performs better than the other two algorithms.

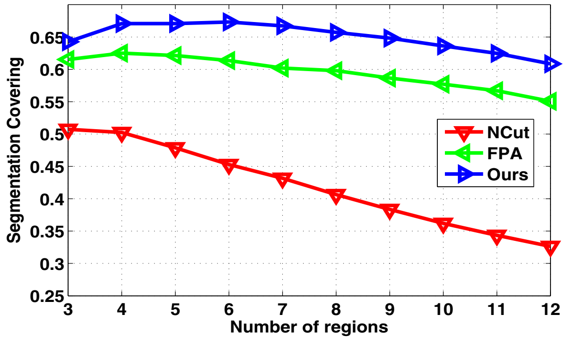

Figure 3. Segmentation covering score of different methods. X axis represents dividing an image into # regions, while Y axis corresponds to the accuracy. NCut means regions generated by normalized cut, FPA means Full Pairwise Affinity model in [8]

Furthermore, as we don't know how many regions should we partition an image into, we consider all the segments generated as a segmentation pool, and recompute the segmentation covering score. The results are shown in Table.I. Again, we can see an obvious advantage of our full model over its counterparts. Besides, the results we achieved is very close to a state-of-art method [26]. However, in [26], it generates a hierarchy of segments of each image, usually it generates hundreds of segments, whilst our results are obtained based only on a total of 75 (3+4+...+12) regions.

Table I
SEGMENTATION COVING SCORE OF THE SEGMENTATION POOL

method	NCut	FPA	Ours	gPb-owt-ucm [26]
Coving	0.60	0.73	0.77	0.78

Fig.1 shows a typical result of our model. It can be seen there that whilst previous methods have failed to recognize the head and the body of the sheep as belonging to one object, our model has succeeded. Some more qualitative

[4] Downloadable from http://www.di.ens.fr/%7Eobozinski/SKMsmo.tar

examples are shown in Fig.4. We can see there that our algorithm can always generate more object-consistent regions than the other segmentation algorithms.

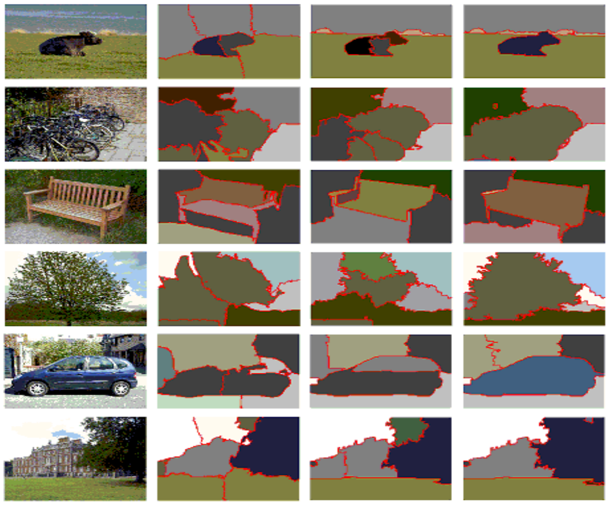

Figure 4. Some examples of the results of three segmentation methods. Each image is segmented into a predefined 7 regions. From left column to right column: original image; results generated by NCuts, Full Pairwise Affinity (FPA) [8], and our full model

B. Comparison of pixel labeling accuracy

Based on the regions generated by our full model, we directly use them as our processing primitives and performed region recognition as detailed in section III-F. Statistics of experimental results are shown in Fig.5 and Table.II. We can see that our full model consistently produced better results than other two methods.

Comparing to most state-of-art results ([2]: 77%, [3]: 76.4%) on this dataset, we are getting very close. Although the hierarchical CRF model of [10] demonstrates superior performance: 86%, it should be noted that their pixel-wise classifier can obtain an overall accuracy of 81%, which suggests their use of much more discriminative features [29].

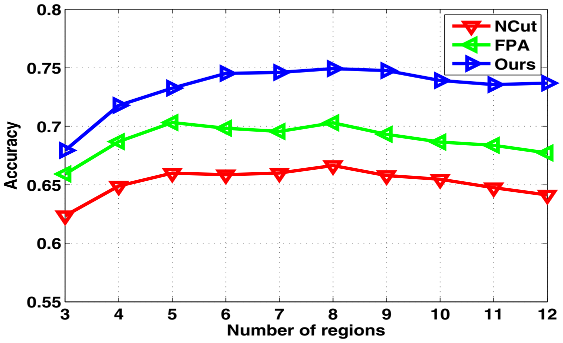

Figure 5. Global pixel accuracy on MSRC21

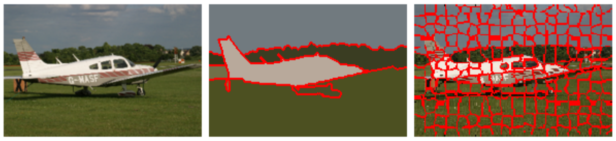

Figure 6. From left to right: original image; object-like regions generated by our full model; superpixels commonly used in conventional semantic segmentation methods

However, the main difference between our algorithm and theirs lies in that their algorithms are superpixel based whilst ours is object-like region based. For a typical image in M-SRC21, it is usually segmented into hundreds of superpixels [3], [2]. However, in our case, we only need to divide the image into a few (3~12) regions and we achieved similar accuracy. It clearly shows the advantages of choosing object-like regions as processing primitives.

Fig.6 shows an illustrative example. It can be seen that whilst superpixel based methods divide the image into many small patches, our model divide it into only a few object like regions. Besides this, our algorithm inherently enjoys the advantage of producing object-consistent outputs as clearly illustrated in Fig.1.

V. Conclusion

In this paper, we first experimentally highlighted the importance of object-level image understanding. To produce the object-like regions, we propose to unify the state-of-art low-level segmentation algorithms with typical semantic segmentation algorithms by introducing the semantic feature feedback. Experiments on MSRC21 confirmed the effectiveness of our new method.

Future works will develop better segmentation methods to produce object-consistent regions. Another problem is how to automatically decide the number of objects contained in an image. In the illustrative example of Fig.1, the appropriate number of regions should be 4 or 6. Dirichlet Process Mixture Model[5] may be a promising tool to deal with this problem.

References

[1] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[2] J. M. Gonfaus, X. Boix, J. V. D. Weijer, A. D. Bagdanov, J. Serrat, and G. Jordi, “Harmony Potentials for Joint Classification and Segmentation,” *CVPR*, 2010.

[3] S. Gould, R. Fulton, and D. Koller, “Decomposing a scene into geometric and semantically consistent regions,” *ICCV*, Sep. 2009.

[5] http://people.csail.mit.edu/jacobe/software/dpmm.tar.gz

Table II
PER-CLASS ACCURACY WHEN EACH TEST IMAGE IS PARTITIONED INTO A FIXED 8 REGIONS

	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat	**Global**	**Average**
[4]	62	98	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	7	72	58
[28]	77	93	70	58	64	92	57	70	61	69	67	74	70	47	80	53	73	53	56	47	40	75	65
NCut	55	86	87	56	27	90	30	62	46	53	60	62	57	24	41	36	72	0	21	48	9	67	49
FPA	61	88	87	63	39	93	36	71	42	43	58	68	49	32	48	45	79	21	32	44	12	70	53
Ours	77	89	85	75	52	80	29	76	70	65	75	65	53	33	71	53	81	44	33	62	10	75	61

[4] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation," *ECCV*, 2006.

[5] T. Malisiewicz and A. A. Efros, "Improving Spatial Support for Objects via Multiple Segmentations," *BMVC*, 2007.

[6] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and their Extent in Image Collections," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, pp. 1605–1614, 2006.

[7] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML*, 2001.

[8] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning Full Pairwise Affinities for Spectral Segmentation," *CVPR*, 2010.

[9] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *ICCV*, vol. 5, no. 6. Citeseer, 2009.

[10] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *2009 IEEE 12th International Conference on Computer Vision*. Ieee, Sep. 2009, pp. 739–746.

[11] J. Carreira and C. Sminchisescu, "Constrained Parametric Min-Cuts for Automatic Object Segmentation," in *CVPR*, 2010.

[12] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[13] V. Lempitsky, A. Blake, and C. Rother, "Image Segmentation by Branch-and-Mincut," in *ECCV*, 2008, pp. 15–29.

[14] B. Alexe, T. Deselaers, and V. Ferrari, "ClassCut for Unsupervised Class Segmentation," in *ECCV*, 2010.

[15] S. Vicente, V. Kolmogorov, and C. Rother, "Cosegmentation Revisited : Models and Optimization," *ECCV*, 2010.

[16] B. Russell, A. Efros, J. Sivic, W. Freeman, and A, "Segmenting Scenes by Matching Image Composites," *NIPS*, 2009.

[17] L.-j. Li, H. Su, E. P. Xing, and L. Fei-fei, "Object Bank : A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification," in *NIPS*, 2010.

[18] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient Object Category Recognition Using Classemes," in *ECCV*, 2010, pp. 776–789.

[19] Z. Tu, "Auto-context and Its Application to High-level Vision Tasks," in *CVPR*, 2008.

[20] S. K. Divvala, A. a. Efros, and M. Hebert, "Can similar scenes help surface layout estimation?" *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2008.

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, no. July, pp. 416–423, 2001.

[22] C. Sutton and A. McCallum, "Piecewise training of undirected models," in *21st Conference on Uncertainty in Artificial Intelligence*. Citeseer, 2005.

[23] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[24] N. Zhou, W. K. Cheung, G. Qiu, and X. Xue, "A Hybrid Probabilistic Model for Unified Collaborative and Content-Based Image Tagging," *PAMI, IEEE Transactions on*, vol. X, no. 99, Nov. 2010.

[25] Y. Jae Lee and K. Grauman, "Object-Graphs for Context-Aware Category Discovery," *CVPR*, 2010.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From Contours to Regions : An Empirical Evaluation," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2294–2301, Jun. 2009.

[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[28] L. Zhang and Q. Ji, "Image segmentation with a unified graphical model," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1406–1425, 2010.

[29] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked Hierarchical Labeling," in *ECCV*, 2010.