A framework for learning to recognize and segment object classes using weakly supervised training data

Caroline Pantofaru and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
crp@ri.cmu.edu, hebert@ri.cmu.edu

Abstract

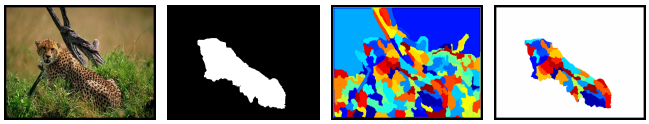
The continual improvement of object recognition systems has resulted in an increased demand for their application to problems which require an exact pixel-level object segmentation. In this paper, we illustrate an example of an object class recognition and segmentation system which is trained using weakly supervised training data, with the goal of examining the influence that different model choices can have on its performance. In order to achieve pixel-level labeling for rigid and deformable objects, we employ regions generated by unsupervised segmentation as the spatial support for our image features, and explore model selection issues related to their representation. Numerical results for pixel-level accuracy are presented on two challenging and varied datasets.

1 Introduction

In this paper, we illustrate an example of an object class recognition system which is trained using weakly supervised training data, and the influence that different model choices can have on its performance. The primary objectives of our system are to *recognize* and *segment* object classes in *novel images* using models learned from *weakly supervised* training data for which image-level labels exist but no indication of individual object localization exists. In order to achieve pixel-level labeling such as in Figure 1 for rigid and deformable objects, we employ regions generated by unsupervised segmentation as the spatial support for our image features. By using established methods for weakly supervised learning, we can concentrate on the actual instantiations of the region representations and examine how different choices for these instantiations affect performance.

Performing pixel-level recognition and segmentation of learned object classes in novel images is itself a very difficult problem which requires image classification to determine the presence of the object, identification of discriminative object parts to roughly localize the object, and identification of non-discriminative object parts to complete the segmentation. For object recognition, local interest point-based methods which use highly discriminative features such as SIFT [17] on fixed-shape object patches have been very successful. These methods do not, however, provide a way to exactly determine which pixels are part of the object unless the shape of the object is rigid or can be parameterized or modelled easily. So we join a growing number of approaches which advocate using regions from unsupervised segmentations to group pixels together and allow proper classification of object parts which are not themselves discriminative [30, 22, 23, 25, 14, 16, 9, 10].

BMVC 2007 doi:10.5244/C.21.30



(a) (b) (c) (d)

Figure 1: For the image of the cheetah in (a), our goal is to produce an exact, pixel-level object segmentation mask as in (b). Regions generated by unsupervised segmentation (c) can be classified to produce such a mask (d).

Many systems which attempt to identify all of the pixels in an object require fully supervised training data that has the objects segmented or extremely obvious [29, 15, 9, 16, 31, 26, 2, 18, 32, 8, 11, 21, 10, 28]. Training data that contains pixel level segmentations is extremely tedious to obtain, however, so such systems are not able to scale to large datasets. On the other hand, object discovery performed in large datasets of completely unlabeled images, such as in [25] can lead to the discovery of objects which are unintuitive or unimportant. Additional constraints may also be needed, for example that an object of interest be exactly denoted by one region [25] or that video data be available [14]. A compromise is to have weakly labeled data in which images are labeled with the objects of interest that they contain, but without the object locations. This type of data is far cheaper to obtain, and has been used in numerous image classification and object recognition tasks that utilize interest points (sparse or dense) or fixed-shape image patches, such as [22]. Very few systems, however, have attempted to use weakly supervised training data and regions to learn object segmentations and object classifiers [1, 6, 3].

As a consequence of using weakly supervised training data and requiring pixel-level object segmentation as output, the image features employed must be carefully designed. As discussed, we cannot use interest points or fixed-shape patches, so we choose to use region-based features. Since we do not know the object shape a priori, we cannot use it as a cue as done in [29, 14, 16]. As argued by [24], we also cannot assume that one segmentation region will correspond to an entire object of interest. So we use and compare three region descriptors which are based on region texture and discriminative region features. Two of the features are adaptations of the texton histograms introduced in [19] to regions, and the third is the Region-based Context Feature (RCF) [23], which includes information about discriminative image features in and around a region. Although other representations are possible, by instantiating our algorithm with these three concrete representations we can study how model changes affect performance.

The structure of the remainder of the paper is as follows. We begin by discussing our system framework which includes our image segmentation method and the learning algorithm used for classifying regions. We then describe the datasets on which results will be presented. With this general framework in mind, we classify single regions using the various representations and compare their performance. We then examine methods for combining representations. Finally, we examine whether incorporating information from neighbouring regions allows for more robust detection and localization.

2 Framework

The general framework for all of the object recognition and segmentation experiments in this paper is as follows: images are first divided into regions using unsupervised seg-

mentation, a representation is then computed for the regions, a model is learned which determines whether a given region feature is indicative of the object in question, and finally novel regions are classified to produce object masks in novel images. This section lays the groundwork for the presented experiments by briefly describing the segmentation and classification methods used, as well as the datasets.

2.1 Segmentation

Although our approach is independent of the segmentation algorithm, it is useful to understand the nature of the regions we are working with. We use mean shift-based segmentation [4] with pixel features of image position (2-dimensional), L*u*v* colour (3-dimensional), and a texton histogram (30-dimensional). An example segmentation is given in Figure 1. We use the algorithm from the Berkeley segmentation database website [19, 20] to compute the texton histograms; the texton at each pixel is a vector of responses to 24 filters quantized into 30 textons, and the texton histogram centred at a pixel is an accumulation of the textons in a 19x19 pixel window. The low dimensionality of our texton histograms allows for generalization during segmentation, grouping together pixels surrounded by similar but not identical textures. Some scale invariance is achieved by segmenting each image at three different image sizes with the same parameters. For clarity, we derive the formalism in this paper for only one segmentation of each image.

2.2 Classification model

Once we have segmented the image into regions, we can represent each region by a feature F (defined in Section 3). To perform region classification, we require a score for each region-representing feature. The method we use for scoring features in our weakly supervised learning environment was introduced for interest points in [5, 27], and for regions in [23]. We assign a score to each feature which indicates how well it discriminates between the positive (object) and negative (background) classes based on the image labels. Let O indicate the *presence* of the positive object class in an *image*, and $\overline{O}$ the *absence* of the object class in an *image*. Then we can define R as the log likelihood ratio of the object's presence, and score $\tilde{R}$ as our posterior belief in O given F (assuming that $P(O) = P(\overline{O})$):

$$R(F) = \log \frac{P(F|O)}{P(F|\overline{O})} \in (-\infty, \infty) \qquad \tilde{R}(F) = P(O|F) = \frac{P(F|O)}{P(F|O) + P(F|\overline{O})} \in [0,1] \quad (1)$$

Note that $\tilde{R}$ preserves the ordering of R but rescales the scores to lie in $[0,1]$. For R, a score approaching negative infinity implies that F indicates a negative image, a score approaching infinity implies that F indicates a positive image, and a score of 0 indicates that F is uninformative for either class. For $\tilde{R}$, a score of 0 implies a negative image, a score of 1 implies a positive image, and a score of 0.5 is uninformative. Laplace smoothing is performed during numerical evaluation of the probabilities for robustness.

For a novel image, we generate its segmentation, compute a feature for each region, and classify each region based on the learned score $\tilde{R}$. The union of all positively classified pixels (or regions) is the object segmentation.

2.3 Datasets

Creating datasets for weakly supervised learning requires much less effort than fully segmenting all of the objects. There is a trade off, however, in that the images chosen for

the weakly supervised datasets must facilitate tractable learning. Specifically, the backgrounds in the images which contain the object must share features with the images which do not contain the object. In this way, we can learn that image features which are seen in both object and non-object images are in fact part of the background. In addition, since we compute pixel-level accuracy, we require images with segmented objects in the *test* image set (but *not* in the training image set).

We present results on two datasets. The first is the 'Spotted Cats' dataset used in [23], which consists of data from the Corel Image Database. The folder 'Cheetahs, Leopards and Jaguars' serves as the positive class, and the folder 'Backyard Wildlife' the negative class. Each folder contains 100 images, half of which are used for training, and half for testing. The images contain cats in a variety of poses, at a variety of sizes, and in varying numbers.

The second dataset we use is the 'Cars' dataset comprised of images from the PASCAL challenge VOC2006 dataset [7]. For training, we use the cars training and validation images (553 images) as the positive set. As the negative training set we use all of the bicycle and bus training and validation images which do not contain cars (509 images). For testing, we use the cars test set (544 images). The PASCAL VOC2006 dataset is extremely challenging, with a large variety of object sizes, poses, lighting conditions, and occlusion. A large number of the images contain multiple cars. We have produced ground truth object masks for the test set for the purpose of the experiments herein.

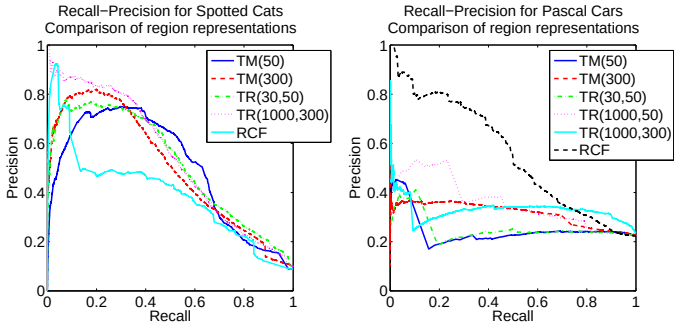
The results we produce are pixel-level, in other words they compute the recall and precision of classifying each individual pixel in each image as object or background. Note that most of the existing evaluations for recognition and localization (which do not rely on human segmentations) are presented as either image classification or object localization in terms of bounding boxes or object centres, with a few exceptions such as [23, 25, 29, 9]. Thus the pixel-level results we present are unusually strict. For most of the experiments, recall-precision curves are presented as in Figure 2. Given the volume of experiments, we summarize some of them in Table 1 by their average precision.

3 Region representations

In order to classify regions we require a way to represent them. In this section, we show that region representation can in fact strongly affect performance. We present three approaches to modeling regions, two based on histograms of textons [19], and another by the Region-based Context Feature [23], and evaluate their performance.

Texture representation 1: mode of the texton histograms in a region. Texture is a representation of the average, or repetitive, patterns within a region. For segmentation, we use a 30-dimensional texton histogram computed at each pixel in the image. Since our segmentation method is mean shift-based, and mean shift computes the mode of the features within a region, a texton histogram mode (TM) for each region results directly from the segmentation. The 30-dimensional texton histogram modes from every training region are clustered into a vocabulary of size $K_{TM} \in \{50, 300\}$, and new modes are assigned to the cluster with the closest centroid.

Texture representation 2: histogram of the textons in a region. As discussed in section 2.1, a low-dimensional texton space facilitates segmentation. However, for region classification a more discriminative texton vocabulary would lead to more discriminative descriptors of region texture. The second texture representation, then, requires computing a new texton vocabulary whose size, $K_{TEX} \in \{30, 200, 1000\}$, is independent of that used for segmentation. The texton words in a segmentation region are then accumulated to



(a) (b)

Figure 2: Comparison of region representations by pixel-level recall-precision curves for recognition and segmentation of (a) Spotted Cats vs. Backyard Wildlife, and (b) PASCAL Cars vs. background. TM means texture representation 1, the texton mode; TR means texture representation 2, the histogram of textons over the region; RCF means Region-based Context Feature. The first number is the size of the texton dictionary used, and the second is the size of the texton histogram mode or texton histogram dictionary.

form a new texton histogram (TR). Note that this texton histogram is computed over the region, not over square windows as in the previous method. These histograms can be clustered to create a different vocabulary of region descriptors of size $K_{TR} \in \{50, 300\}$.

Region-based Context Features. The shape of a segmentation region is data-driven, which is useful for specifying the spatial support of texture features, but it can also work against us. Consider the images in the top row of Figure 4(b,e). The regions on the body of the cheetah are detected using texture, but the head and face are missed since they have little interior texture. The head regions do, however, have discriminative features such as the shape of the ear or the presence of an eye. Cars also have little texture within a region, but the shape of the wheels are strong cues. For these image structures, we require a representation which can both encapsulate their uniqueness and extend past the boundaries of the region to capture shape. To address these concerns, we adopt the Region-based Context Feature (RCF) [23]. RCFs are the cluster centres of clustered histograms of quantized SIFT [17] descriptors computed at local interest points in and *around* a given region. The SIFT descriptors ensure their discriminative power, and the histograms include points outside the region in a principled manner. For implementation details, please see [23].

Experiments. We experiment with model selection for region representation by choosing a representation type, TM, TR or RCF, a texton or SIFT dictionary size K_{TEX}, and a size for the descriptor, or histogram, dictionary K_{TM}, K_{TR} or K_{RCF}. In Figure 2 and Table 1 we compare choices for these parameters. Figure 2 presents recall-precision curves for pixel-level detection on the two datasets. For each pixel in each image, we evaluate whether it has been assigned to the correct class. Curves are shown for the TM representation (texton representation 1) with $K_{TM} \in \{50, 300\}$, the TR representation (texton representation 2) with $(K_{TEX}, K_{TR}) \in \{(30, 50), (1000, 300)\}$, and the RCF representation with $K_{SIFT} = K_{RCF} = 50$. Curves corresponding to the other model choices are omitted for clarity. Table 1 presents values for the average precision of all model choices. The top row corresponds to the representation, the second row to the texton or SIFT dictionary size, and the left column to the feature dictionary size.

While larger dictionaries seem intuitively more desirable, the results in Figure 2 dis-

	TM 30	TR 30	TR 200	TR 1000	RCF 50
50	0.493	0.513	0.504	0.542	0.392
300	0.498	0.493	0.533	0.545	-

(a)

	TM 30	TR 30	TR 200	TR 1000	RCF 50
50	0.246	0.253	0.302	0.353	-
300	0.316	0.261	0.346	0.329	0.548

(b)

Table 1: Comparison of region representations by average precision computed on (a) the Spotted Cats dataset, and (b) the PASCAL cars dataset. The first row indicates the region representation used, the second row indicates the size of the texton (or SIFT) dictionary, and the left column indicates the size of the descriptor (histogram) dictionary. Note that the average precisions do not increase monotonically with dictionary size.

agree. For the TM representation on the Spotted Cats dataset, a dictionary size of 300 performs better for low recall values, a dictionary size of 50 performs better for medium recall values, and they perform similarly for high recall values. This trend can also be seen for the TR curves for the Spotted Cats. In Table 1 we can see that the average precision does not increase monotonically with dictionary size, reinforcing the fact that model selection is crucial to meeting specific system requirements.

4 Combining features

In the previous section, we described three different ways to represent segmentation-generated regions for object recognition and localization: TM, TR and RCFs. We found that the texture descriptors are better suited to objects with regular texture, such as spotted cats, while the RCFs are better suited to objects with discriminative parts, such as cars. This section focuses on combining the representations with the goal of eliminating the need to make a class-dependent choice between them.

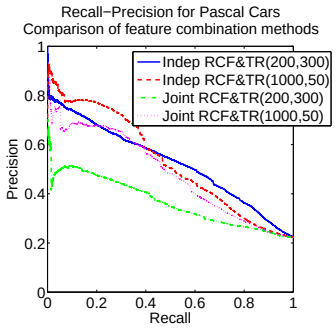


Figure 3: Recall-precision curves comparing feature combination methods for pixel-level recognition and segmentation of the PASCAL Cars. 'Indep' means combining features assuming independence; 'Joint' means using joint distributions.

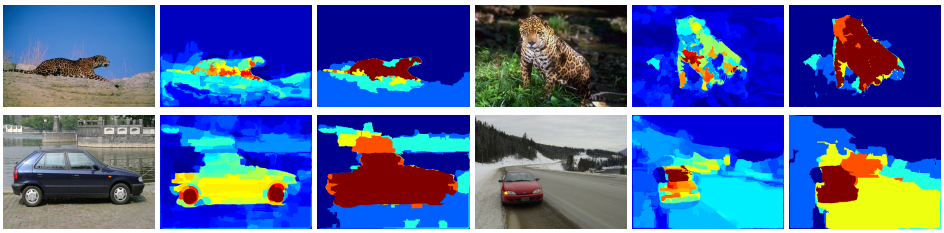
To combine feature sets, a simplifying assumption that could be made is feature independence. Although a texture representation of a region and an RCF are clearly not independent, this naïve Bayes assumption allows us to estimate their joint probability from relatively little data. We define the independent score for a feature pair to be:

$$\tilde{R}(T,RCF) \propto \tilde{R}(T)\tilde{R}(RCF)$$

Where T is a texture feature, and RCF is a Region-based Context Feature.

With sufficient training data, the score of the joint probability of the texture features and RCFs, $\tilde{Rj}(T,RCF)$, can be modeled as for a single feature:

$$Rj(T,RCF) = \log \frac{P(T,RCF|O)}{P(T,RCF|\bar{O})} \qquad \tilde{Rj}(T,RCF) = \frac{P(T,RCF|O)}{P(T,RCF|O)+P(T,RCF|\bar{O})}$$



(a) (b) (c) (d) (e) (f)

Figure 4: Examples of results on the Spotted Cats and Cars datasets. Columns (a) and (d) show the original images, (b) and (e) shown the results of using single region classification with feature TR(1000,300) for the Spotted Cats and RCFs for the Cars, and (c) and (f) show the results of incorporating neighbourhood information. Results are in the 'jet' colourmap, ranging from dark blue for background to dark red for object detections.

In Figure 3 we compare the effects of using the two feature combination models. For clarity, only two representative curves per model are plotted. The combination of RCFs with the TR texture representation with $(K_{TEX}, K_{TR}) = (1000, 50)$ provided the best results for both combination methods, surpassing even $(K_{TEX}, K_{TR}) = (1000, 300)$. Despite a large amount of training data, the best results are not always obtained with the largest dictionaries. Another counter-intuitive result is that combining the features with an assumption of independence performs better than modeling the joint distribution even on the Cars dataset which contains more than 1000 training images. Finally, note that although the combination $(K_{TEX}, K_{TR}) = (200, 300)$ performs very poorly in the joint distribution representation, it outperforms $(K_{TEX}, K_{TR}) = (1000, 50)$ for high recall values in the independent distribution representation. This indicates that performance under one type of distribution does not mirror performance under the other. This result can be applied to other object recognition systems, suggesting that many common assumptions in system implementations should in fact be verified.

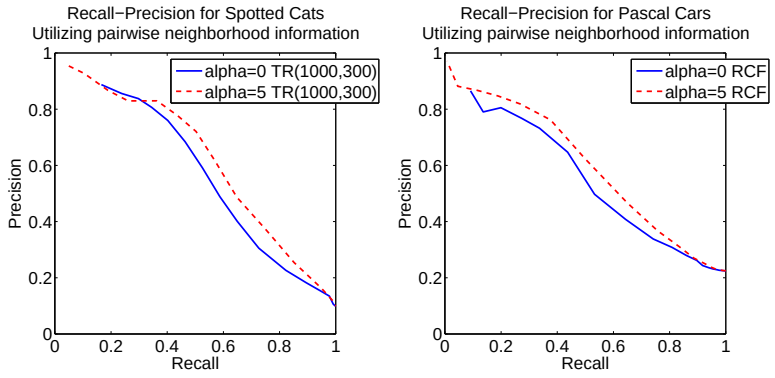
5 Incorporating neighbourhood information

We have made progress in classifying regions to detect and segment objects, however we still make mistakes, especially with respect to spurious object detections and missed detections which are inconsistent with the regions around them, such as in columns (b) and (e) of Figure 4. In this section we describe a method for smoothing the localization results using information from adjacent pairs of regions which leads to the results in columns (c) and (f) of Figure 4.

To incorporate pairwise information, the two-class classification problem can be reformulated as an energy minimization. Let $x_i \in \{0, 1\}$ be the label of region i, with 0 for background and 1 for object. Then $\mathbf{X} = [x_1..x_N]$ is the vector of assignments to every region in an image. Also, let F_i be the feature cluster for region i, and $\mathbf{F} = [F_1..F_N]$ be the vector of region features over the entire image. Here F can be a texture feature or an RCF. We define the energy of an assignment of labels X to an image to be:

$$E(\mathbf{X}, \mathbf{F}) = \sum_i U(x_i, F_i) + \alpha \sum_i \sum_j B(x_i, x_j, F_i, F_j) \quad (2)$$

Provided that the pairwise term $B(x_i, x_j, F_i, F_j)$ is associative, we can use graph cuts to



(a) (b)

Figure 5: Pixel-level precision-recall curves for the (a) Spotted Cats and (b) Cars datasets using pairwise neighbourhood information. The best-performing single-region representations were used, TR(1000,300) for the Spotted Cats and RCFs for the Cars. Each curve shows the results using a different α as in Equation 2. For $\alpha = 0$ the binary energy term is ignored, and the results mirror those of single regions in Figure 2. From the $\alpha = 5$ curve we see that incorporating neighbourhood information can indeed improve precision by 5-10% for many of the recall values.

minimize the energy exactly and obtain a region labeling [13].

In the energy function, the unary term serves to classify single regions. By using a generative model, we can define the unary energy term as:

$$-\log P(F_i|x_i) = -\log P(F_i|x_i=0)^{(1-x_i)} P(F_i|x_i=1)^{x_i} = -x_i \log \frac{P(F_i|x_i=1)}{P(F_i|x_i=0)} - \log P(F_i|x_i=0)$$

Note that $\log P(F_i|x_i=0)$ is constant with respect to the label assigned to x_i, so we can ignore it in the energy minimization. Also, as in the single-region classifier, we can use the bounded posterior $\tilde{R}(F_i)$ instead of the unbounded log likelihood ratio. Note that this is an approximation computed using *image* labels, not *region* labels. Finally, energy minimization via graph cuts produces a binary labeling of the data, but it does not produce marginals. To compute different recall-precision values, we use a penalty term β which adjusts the threshold between declaring a region part of an object or the background. Another method for achieving this goal is presented in [12]. Our final unary term is:

$$U(x_i, F_i) = -x_i \left(\tilde{R}(F_i) - \beta\right)$$

In most energy minimizations, the binary energy term is a classifier between the four possible assignments $(x_i, x_j) \in \{(0,0),(0,1),(1,0),(1,1)\}$ trained using fully labeled data. Since we only have image-level labels, defining and training the binary energy term is a more difficult process. We choose to define $B(x_i, x_j, F_i, F_j)$ as a smoothing term which only differentiates between the two regions having the same label or different labels. The lack of training data with adjacent neighbours labeled $(0,1)$ or $(1,0)$ makes this impossible to train directly, so we define the binary terms as:

$$B(x_i, x_j, F_i, F_j) \quad = \quad \begin{cases} log(1 + \frac{P(F_i, F_j | i \bowtie j)}{P(F_i, F* | i \bowtie j) + P(F*, F_j | i \bowtie j)}) & x_i \neq x_j \\ 0 & x_i = x_j \end{cases} \tag{3}$$

Where $i \bowtie j$ means that i and j are neighbours, and $F*$ is any feature cluster. The $1+$ term ensures that $B >= 0$ and hence is associative. By this definition of B, we prefer to smooth region labels if the regions are frequently seen together and hence are likely to belong to the same object. Features which are rarely seen together are more likely to belong to different objects, hence their respective region labels are less prone to smoothing.

We performed experiments using this energy function on both datasets. The features used were the TR(1000,300) for the Spotted Cats, and the RCFs for the Cars, since they performed best on the single-region representation task. Values of α of 0, 0.5, 1 and 5 were used. In Figure 5 we show the results of setting $\alpha = 0$ and $\alpha = 5$. We omit the $\alpha = 0.5$ and $\alpha = 1$ settings for clarity since they lie strictly between the $\alpha = 0$ and $\alpha = 5$ curves. If $\alpha = 0$, the binary term in the energy function is ignored, and indeed this curve matches the single-representation RCF curve in Figure 2.

From the $\alpha = 5$ curve we can see that using this pair-wise smoothing approach does indeed improve precision by 5-10% for most recall values. Example results are given in Figure 4.

6 Conclusions

In this paper we have presented a study of issues related to using segmentation-generated regions for object class recognition and segmentation in a weakly supervised learning framework through a particular instantiation of such a system. Results were computed for pixel-level localization. We explored three different region representations and model selection issues for each, concluding that the choices are dataset-dependent and hence multiple representations should be evaluated for future systems in this category.

In order to reduce the burden of region representation, we examined ways to combine representations that would perform at least as well as a single representation alone. Our results showed that modeling the joint distribution of the representations by assuming their independence can lead to promising performance. However, even a large dataset such as the 1000 training images we used for the PASCAL Cars was not enough to properly model a full joint distribution between representations.

Finally, we attempted to reduce the spurious object detections and spurious object misses by incorporating information from a region's neighbourhood. Both single regions and pairs of neighbouring regions were used to compute an energy-minimizing labeling solution which outperformed the use of single regions alone.

References

[1] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *CVPR*, 2003.

[2] N. W. Campbell, B. T. Thomas, and T. Troscianko. A two-stage process for accurate image segmentation. In *Int. Conf. on Image Processing and its Applications*, 1997.

[3] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[5] G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, volume 1, pages 634–640, 2003.

[6] P. Duygulu, K. Barnard, N. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[7] M. Everingham, L. V. Gool, C. Williams, and A. Zisserman. The pascal 2006 visual object classes challenge.

[8] M. R. Everingham, B. T. Thomas, and T. Troscianko. Head-mounted mobility aid for low vision using scene classification techniques. *Int'l Journal of Virtual Reality*, 3, 1999.

[9] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006.

[10] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proceedings of ICCV*, 2005.

[11] I. Y. Kim and H. S. Yang. Efficient image understanding based on the markov random field model and error backpropagation network. *PAMI*, 1992.

[12] P. Kohli and P. Torr. Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts. In *ECCV*, 2006.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002.

[14] M. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[15] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.

[16] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] W. P. J. Mackeown, P. Greenway, B. T. Thomas, and W. A. Wright. Labelling images with a neural network. In *ICANN*, 1993.

[19] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue combination in image segmentation. In *ICCV*, 1999.

[20] 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. In *ICCV*, 2001.

[21] J. W. Modestino and J. Zhang. A markov random field model-based approach to image interpretation. *PAMI*, 14, 1992.

[22] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *SCIA*, 2005.

[23] C. Pantofaru, G. Dorkó, C. Schmid, and M. Hebert. Combining regions and patches for object class localization. In *Beyond Patches Workshop, CVPR*, 2006.

[24] C. Pantofaru and M. Hebert. A comparison of image segmentation algorithms. Technical Report CMU-RI-TR-05-40, Robotics Institute, Carnegie Mellon University, September 2005.

[25] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of CVPR*, June 2006.

[26] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *CVPR*, 2006.

[27] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, 2001.

[28] F. Schroff, A. Criminisi, and A. Zisserman. Single-histogram class models for image segmentation. In *ICVGIP*, 2006.

[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[30] Z. Tu, Z. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005.

[31] F. Vivarelli and C. Williams. Using bayesian neural networks to classify segmented images. Technical report, Aston University, 1997.

[32] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006.