Using Multiple Segmentations to Discover Objects and their Extent in Image Collections

Bryan C. Russell[1] Alexei A. Efros[2] Josef Sivic[3] William T. Freeman[1] Andrew Zisserman[3]

[1] CS and AI Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, U.S.A.
{brussell,billf}@csail.mit.edu

[2] School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
efros@cs.cmu.edu

[3] Dept. of Engineering Science
University of Oxford
Oxford, OX1 3PJ, U.K.
{josef,az}@robots.ox.ac.uk

Abstract

Given a large dataset of images, we seek to automatically determine the visually similar object and scene classes together with their image segmentation. To achieve this we combine two ideas: (i) that a set of segmented objects can be partitioned into visual object classes using topic discovery models from statistical text analysis; and (ii) that visual object classes can be used to assess the accuracy of a segmentation. To tie these ideas together we compute multiple segmentations of each image and then: (i) learn the object classes; and (ii) choose the correct segmentations. We demonstrate that such an algorithm succeeds in automatically discovering many familiar objects in a variety of image datasets, including those from Caltech, MSRC and LabelMe.

1. Introduction

In [21] we posed the question, given a (Gargantuan) number of images, "Is it possible to learn visual object classes simply from looking at images?". That is, if our data set contains many instances of (visually similar) object classes, can we *discover* these object classes? In this paper we extend this question to "Is it possible to learn visual object classes *and their segmentations* simply from looking at images?"

To automatically discover objects in an image collection, two very challenging issues must be addressed: (i) how to recognize visually similar objects; and (ii) how to segment them from their background. But, in a sense, both object recognition and image segmentation can be thought of as parts of one large *grouping problem* within the space of an entire dataset. Given a stack of all images in the dataset, groups representing similar objects can be seen as volumes in that stack. Projecting such volumes onto a particular image gives segmentation; projecting onto the image index gives recognition. Our aim here is to couple object-based matching/recognition and image-based segmentation into a general grouping framework.

To be concrete, the problem that we wish to solve is the following: given a large dataset of images (containing multiple instances of several object classes), retrieve segmented instances grouped into object classes. The hope is that this will recover commonly occurring object classes in the dataset (e.g. cars, buildings). Our approach is to first obtain multiple segmentations of each image, and to make the assumption that each object instance is correctly segmented by at least one segmentation. The problem is then reduced to finding coherent groups of correctly segmented objects within this large "soup" of candidate segments, i.e. one of grouping in the space of candidate image segments. Our approach is illustrated in figure 1.

1.1. Background

Several researchers have proposed mining large visual datasets to cluster multiple *instances* of objects. Examples include discovering main characters [10] and other prominent objects and scenes [23] in movies or mining famous people in collections of news photographs [1]. Recently, some success has also been reported in discovering object and scene *categories* [7, 17, 21] by borrowing tools from the statistical text analysis community. These tools, such as probabilistic Latent Semantic Analysis (pLSA) [12] and Latent Dirichlet Allocation (LDA) [2], use unordered "bag of words" representation of documents to automatically discover topics in a large text corpus. To map these techniques onto the visual domain, an equivalent notion of a text word needs to be defined. Most researchers follow the approach of using clustered affine-invariant point descriptors as "visual words" [5, 22]. Under this model, images are treated as documents, with each image being represented by a histogram of visual words. Applying topic discovery to such a representation is successful in classifying the image, but the resulting object segmentations are "soft" – the discovered objects (or scenes) are shown by highlighting the visual

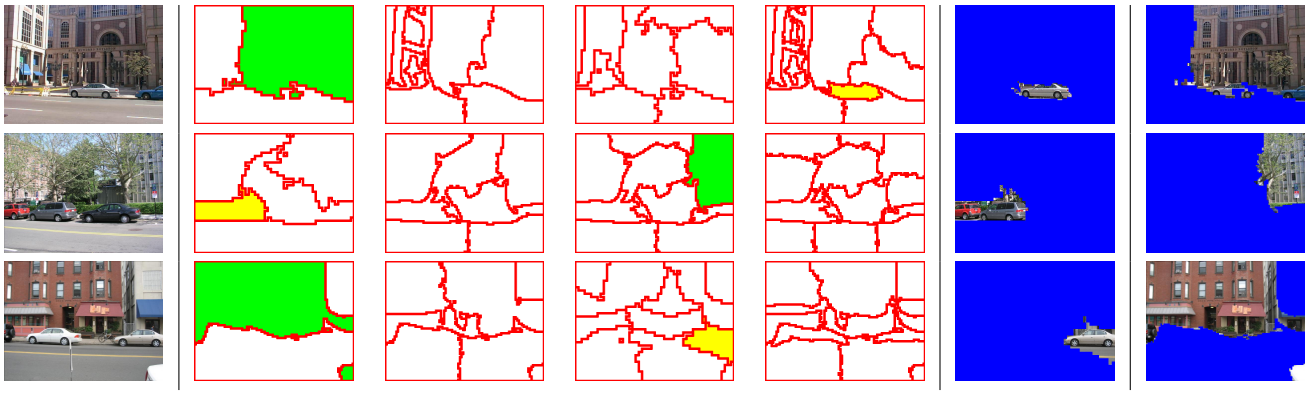


Figure 1. **Problem summary**. Given a set of input images (first column), we wish to discover object categories and infer their spatial extent (e.g. cars and buildings: final two columns). We compute multiple segmentations per image (a subset is depicted in the second through fifth columns; all of the segmentations for the first row are shown in Figure 4). The task is to sift the good segments from the bad ones for each discovered object category. Here, the segments chosen by our method are shown in green (buildings) and yellow (cars).

words in the image belonging to a particular topic.

One major issue noticed by several groups [17, 21], is that the "visual words" are not always as descriptive as their text counterparts. While some visual words do capture high-level object parts, (e.g. wheels, eyes, airplane wingtips), many others end up encoding simple oriented bars and corners and might more appropriately be called "visual phonemes" or even "visual letters". Consequently, there is a proportion of visual synonyms – several words describing the same object or object part, and, more problematically, visual polysemy – the same word describing several different objects or object parts. All this means that the statistical text methods alone are sometimes not powerful enough to deal with the visual data. This is not too surprising – after all, the visual world is much richer and noisier than the human-constructed, virtually noiseless world of text.

1.2. Grouping visual words

The problem of visual polysemy becomes apparent when we consider how an image is represented in the "bag of words" document model. All visual words in an image are placed into a single histogram, losing all spatial and neighborhood relationships. Suppose a car is described by ten visual words. Does the presence of these ten words in an image imply that it contains a car? Not necessarily, since these ten words did not have to occur together spatially, but anywhere in the image. Of course, if the object and its background are highly correlated (e.g. cars and roads or cows and grass), then modeling the entire image can actually help recognition. However, this is unlikely to scale as we look at a large number of object classes. Therefore, what we need is a way to group visual words spatially [8, 24] to make them more descriptive.

1.3. Multiple segmentation approach

In this paper we propose to use image segmentation as a way to utilize visual grouping cues to produce groups of related visual words. In theory, the idea sounds simple: compute a segmentation of each image so that each segment corresponds to a coherent object. Then cluster similar segments together using the "bag of words" representation. However, image segmentation is not a solved problem. It is naive to expect a segmentation algorithm to partition an image into its constituent objects – in the general case, you need to have solved the recognition problem already! In practice, some approaches, like Mean-shift [4], perform only a low-level over-segmentation of the image (superpixels). Others, like Normalized Cuts [20] attempt to find a global solution, but often without success (however, see Duygulu *et al.* [6] for a clever joint use of segments and textual annotations).

Recently, Hoiem *et al.* [13] have proposed a surprisingly effective way of utilizing image segmentation without suffering from its shortcomings. For each image, they compute *multiple* segmentations by varying the parameters of the segmenting algorithm. Each of the resulting segmentations is still assumed to be wrong – but the hope is that *some* segments in *some* of the segmentations will be correct. For example, consider the images in figures 1 and 4. None of the segmentations are entirely correct, but most objects get segmented correctly at least once. This idea of maintaining multiple segmentations until further evidence can be used to disambiguate is similar to the approach of Borenstein *et al.* [3].

The problem now becomes one of going through a large "soup" of (overlapping) segments and trying to discover the good ones. But note that, in a large image dataset with many examples of the same object, the good segments (i.e. the ones containing the object) will all be represented by a similar set of visual words. The bad segments, on the other hand, will be described by a random mixture of object-words and background-words. To paraphrase Leo Tolstoy [25]: *all good segments are alike, each bad segment is bad in its own way*. This is the main insight of the paper: segments cor-

Given a large, unlabeled collection of images:

1. For each image in the collection, compute multiple candidate segmentations, e.g. using Normalized Cuts [20] (section 2.1).
2. For each segment in each segmentation, compute a histogram of "visual words" [22] (section 2.2).
3. Perform topic discovery on the set of *all segments* in the image collection (using Latent Dirichlet Allocation [2]), treating each segment as a document (section 2.3).
4. For each discovered topic, sort *all segments* by how well they are explained by this topic (section 2.4).

Figure 2. Algorithm overview.

responding to objects will be exactly the ones represented by coherent groups (topics), whereas segments overlapping object boundaries will need to be explained by a mixture of several groups (topics). We exploit this insight in the object discovery algorithm described next.

2. The Algorithm

Given a large, unlabeled collection of images, our goal is to automatically discover object categories with the objects segmented out from the background. Our algorithm is summarized in figure 2.

The result is a set of discovered topics, where the top-ranked discovered segments correspond to the objects within that topic. The rest of the section will describe the steps of the algorithm in detail.

2.1. Generating multiple segmentations

Our aim is to produce sufficient segmentations of each input image to have a high chance of obtaining a few "good" segments that will contain potential objects. There are approaches in the literature for sampling likely segmentations [26] and multiscale segmentations [19]. But since we are not relying on the full segmentation to be correct, the particular choice of a segmentation algorithm is not that critical. Indeed, the fact that segmentation algorithms are not particularly stable as one perturbs their parameters is exactly what we use to obtain a variety of different segmentations.

We have chosen the Normalized Cuts framework [20], because it aims to produce a global segmentation with large segments that have a chance to be objects. The affinity metric we use is the intervening contour cue based on the texture-suppressing boundary detector of Martin *et al.* [16]. To produce multiple segmentations, we varied two parameters of the algorithm: the number of segments K and the size of the input image. We typically set $K = 3, 5, 7, 9$ segments and applied these settings at 2 image scales: 50- and 100-pixels across (for the LabelMe dataset, we also used $K = 11, 13$ and for the MSRC dataset we added a third scale at 150-pixels across). This results in up to 12 different segmentations per image, for a total of up to 96 (overlapping) segments. Figure 4 shows the set of resulting segmentations for sample images.

2.2. Obtaining visual words

The goal is to develop a description of an image segment which would have tolerance to intra-class variations and a certain degree of viewpoint and lighting changes. Due to imperfections in segmentation the representation should be also tolerant to some amount of partial occlusion and clutter, e.g. a segment containing a 'car' might have a roof missing and/or include a part of the road.

We follow the approach of [21] and represent images using affine covariant regions, described by SIFT [15] descriptors and quantized into approximately 2,000 visual words. The regions are computed using binaries provided at [14]. The quantization is performed by k-means clustering of regions from 1,861 images of cars, faces, motorbikes, airplanes and backgrounds from the Caltech dataset [9]. Note that the same cluster centers (visual words) are used for all experiments in this paper.

Once the visual words are computed for an image, each image segment is represented by a histogram of visual words contained within the segment (the bag of words model).

2.3. The topic discovery models

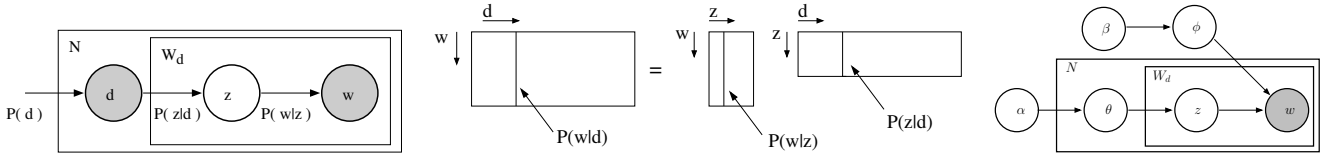
We review the topic discovery models from statistical text analysis, Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA), which we apply here in the visual domain. The goal is to analyze the collection of segments and discover 'topics', which should correspond to visually similar objects frequently occurring in the data.

We will describe the models using the original terms 'documents' and 'words' as used in the text literature. In our case, documents correspond to image segments (section 2.1) and words correspond to quantized affine covariant regions (section 2.2).

Suppose we have N documents containing words from a vocabulary of size M. The corpus of text documents is summarized in a M by N co-occurrence table $\mathtt{N}$, where $n(w_i, d_j)$ stores the number of occurrences of a word w_i in document d_j. In addition, there is a hidden (latent) topic variable z_k associated with each occurrence of a word w_i in a document d_j.

The joint probability $P(w_i, d_j, z_k)$ is assumed to have the form of the graphical model shown in figure 3(a). Marginalizing over topics z_k determines the conditional probability $P(w_i|d_j)$:

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k), \qquad (1)$$



(a) (b) (c)

Figure 3. (a) pLSA graphical model, see text. Nodes inside a given box (plate notation) indicate that they are replicated the number of times indicated in the top left corner. Filled circles indicate observed random variables; unfilled are unobserved. (b) In pLSA the goal is to find the topic specific word distributions $P(w|z_k)$ and corresponding document specific mixing proportions $P(z|d_j)$ which make up the document specific word distribution $P(w|d_j)$. (c) LDA graphical model.

where $P(z_k|d_j)$ is the probability of topic z_k occurring in document d_j; and $P(w_i|z_k)$ is the probability of word w_i occurring in a particular topic z_k.

The model (1) expresses each document as a convex combination of K topic vectors. This amounts to a matrix decomposition as shown in figure 3(b) with the constraint that both the vectors and mixture coefficients are normalized to make them probability distributions. Essentially, each document is modeled as a mixture of topics – the histogram for a particular document being composed from a mixture of the histograms corresponding to each topic.

In contrast to pLSA, LDA treats the multinomial weights $P(z|d)$ over topics as latent random variables. The pLSA model is extended by sampling those weights from a Dirichlet distribution, the conjugate prior to the multinomial distribution. This extension allows the model to assign probabilities to data outside the training corpus and uses fewer parameters, thus reducing overfitting (see [2] for a detailed comparison). The LDA model is shown in Figure 3(c), where W_d is the number of words in document d. The goal is to maximize the following likelihood:

$$p(\boldsymbol{w}|\phi,\alpha,\beta) = \int \sum_{\boldsymbol{z}} p(\boldsymbol{w}|\boldsymbol{z},\phi)p(\boldsymbol{z}|\theta)p(\theta|\alpha)p(\phi|\beta)d\theta \tag{2}$$

where θ and ϕ are multinomial parameters over the topics and words respectively and $p(\theta|\alpha)$ and $p(\phi|\beta)$ are Dirichlet distributions parameterized by the hyperparameters α and β. Since the integral is intractable to solve directly, we solve for the ϕ parameters using Gibbs sampling, as described in [11]. We ran the Gibbs sampler for 100 iterations, which converged on a Pentium 2.2 GHz machine in under 2 hours on the MSRC dataset with approximately 200K segments.

The hyperparameters control the mixing of the multinomial weights (lower values give less mixing) and can prevent degeneracy. As in [11], we specialize to scalar hyperparameters (e.g. $\alpha_i = a \; \forall \; i$). For this paper, we used $\alpha_i = 0.5$ and $\beta_j = 0.5$.

2.4. Sorting the soup of segments

We wish to find good segments within each topic. We sort the segments by the similarity of the visual word distribution (normalized histogram) within each segment to the learned multinomial weights ϕ_t for a given topic t. Let ϕ_s be the multinomial parameter describing the visual word distribution within a segment. We sort the segments based on the Kullback-Leibler (KL) divergence $D(p(w|s,\phi_s)||p(w|z,\phi_t))$ between the two distributions over visual words.

Figure 4 shows discovered objects segmented out of the image. We also show the generated multiple segmentations and have weighted each segment based on their KL divergence score. Notice that often there is a tight segmentation of the discovered objects.

3. Results

In this section, we show qualitative results on several datasets and report quantitative results on two tasks: (i) the retrieval task, where we wish to evaluate whether or not the top ranked images for a particular topic contain the discovered object; and (ii) the segmentation task, where we wish to evaluate the quality of object segmentation and the proportion of well-segmented highly-ranked objects.

Image datasets: We investigated three datasets: Caltech [9], MSRC [27], and LabelMe [18]. A summary of the object categories and number of images used appears in table 1. We tested on progressively more difficult datasets. For the Caltech set, we used four object categories – the 'Caltech four' of [9] – each containing a single instance appearing in flat or cluttered background, and a set of background images. The MSRC set contains 23 object and scene categories. Many of the objects in this set are prominently featured and located close to the center of the image. There are also many images containing multiple objects, with some that are occluded. The LabelMe dataset is a more difficult collection of scene images where the objects are not controlled and appear in their natural habitat. For this set, we queried for images containing cars, trees, and buildings. The query resulted in 1554 images, containing many other additional objects.

Figures 5-7 show montages of segments for each topic, sorted by their KL divergence score. Note that for each discovered category, the objects are reasonably segmented and are consistent. The depicted segments each come from a

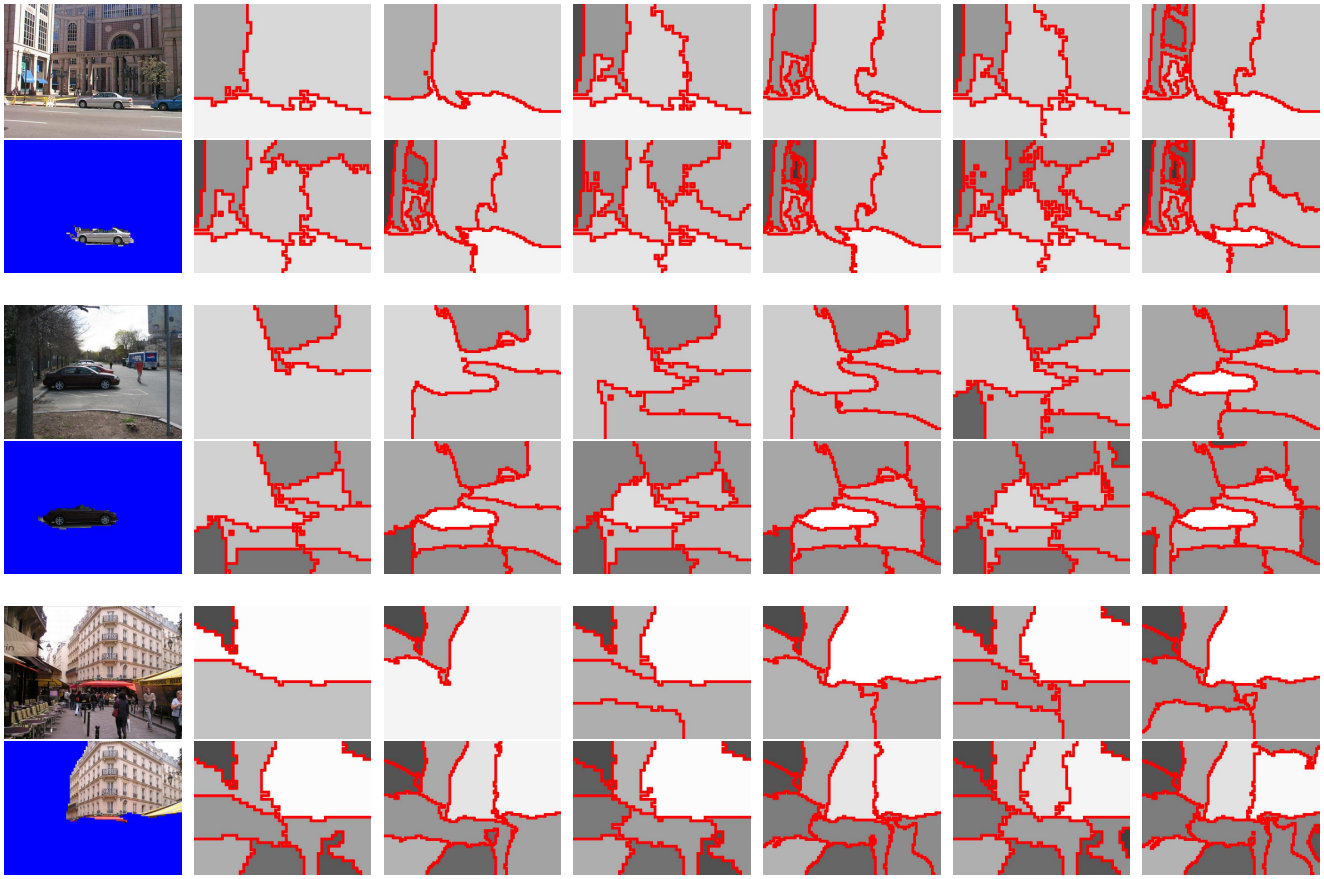


Figure 4. How multiple candidate segmentations are used for object discovery. The top left image of every pair of rows is the input image, which is segmented using Ncuts at different parameter settings into 12 different sets of candidate regions. The explanatory power of each candidate region is evaluated as described in the text; we illustrate the resulting rank by the brightness of each region. The image data of the top-ranked candidate region is shown in the bottom left, confirming that the top-ranked regions usually correspond to objects.

different image to avoid showing multiple segments of the same object.

To assess the contributions of the different steps of the algorithm, we evaluate: (a) the proposed algorithm (of figure 2), (b) swapping the LDA model for the simpler pLSA model to evaluate the contribution of the Dirichlet prior over the multinomial weights, (c) using only a single segmentation for each image (in conjunction with the LDA model) to evaluate the contribution of computing multiple segmentations for each image, (d) our previous method [21], where we use no segmentation at all and each image is treated as a separate document, with the object extent determined by the union of visual words having high posterior probability (greater than 0.5) for a particular topic. For all tests, each method was run 10 times and the run with the highest likelihood was used.

Image retrieval performance is evaluated on the MSRC database, where labels indicating object presence/absence are available. The evaluation is performed for four objects: 'bicycles', 'cars', 'signs' and 'windows'. For the proposed method (a), top ranked images for corresponding topics are shown in figure 7. Precision-recall curves were computed and the average precision is reported in table 2 for the tested methods.

For 'bicycles' and 'windows', the proposed method performs on par or better than the other methods. Method (d), where no segmentation is used, performs best on 'cars' because it is learning about other objects in the scene that overlap significantly with the target object (e.g. roads). These other objects predict well the presence and location of the target object for the tested dataset. This effect may also explain why method (c), which uses a coarse segmentation, performs better on 'signs'. Method (b) performs significantly worse than the other methods. We believe this is due to pLSA overfitting the data, because of the lack of a Dirichlet prior on the document-topic coefficients [2]. In our earlier work [21], we did not observe a significant difference in performance between pLSA and LDA. This might be due to the smaller number of topics and documents used. Our earlier work had only about 4K documents and 4-7 topics, whereas in this work we have about 200K documents and 25 topics.

Dataset	# of images	# of categories
Caltech [9]	4,090	4 + background
MSRC [27]	4,325	23 object and scene categories
LabelMe [18]	1,554	cars, buildings, trees

Table 1. Summary of datasets used in this paper.

Method	bicycles	cars	signs	windows
(a) Mult. seg. LDA	0.69	0.77	0.43	0.74
(b) Mult. seg. pLSA	0.67	0.28	0.34	0.57
(c) Sing. seg. LDA	0.67	0.73	0.46	0.72
(d) No seg. LDA	0.64	0.85	0.40	0.74
(e) Chance	0.06	0.12	0.04	0.15

Table 2. Average precisions for the tested methods on several objects from the MSRC dataset.

Method	buildings	cars	roads	sky
(a) Mult. seg. LDA	0.53	0.21	0.41	0.77
(b) Mult. seg. pLSA	0.59	0.09	0.16	0.77
(c) Sing. seg. LDA	0.55	0.29	0.32	0.65
(d) No. seg. LDA	0.47	0.16	0.14	0.16

Table 3. Segmentation score for the tested methods on several objects with ground truth labels from the LabelMe dataset. See text for a description of the segmentation score.

The segmentation accuracy is evaluated on the LabelMe dataset, where ground truth object segmentation was labelled for each tested method on the top twenty returned images for topics covering four objects: 'buildings', 'cars', 'roads' and 'sky'. Let R and GT be respectively the set of pixels in the retrieved object segment and the ground truth segmentation of the object. The performance score ρ measures the area correctly segmented by the retrieved object segment. It is the ratio of the intersection of GT and R to the union of GT and R, i.e. $\rho = \frac{GT \cap R}{GT \cup R}$. If more than one ground truth segmentation intersects R, then we use the one that results in the highest score. The score is then averaged over the top 20 retrieved object segments. The results are summarized in table 3.

Our method scores about the same or better than the other methods on 'roads' and 'sky' objects. Methods (b) and (c) perform better on 'building' and 'car' objects respectively. Note that this comparison takes into account only the top 20 segments for each method and does not measure the number of top-ranked high quality segments. For the 'car' object, we have closely inspected the results of methods (a) and (c). While the quality of segmentations is worse in the top 20 returned images, the proposed method (a) outperforms single segmentation LDA (c) over the top 500 returned images (the proposed method returns about 15% more high quality segments). This suggests that using multiple segmentations generates more high quality segments in the dataset.

4. Conclusion

By combining multiple candidate segmentations with probabilistic document analysis methods, we have developed an algorithm that finds and segments visual topics within an unlabeled collection of images. The discovered topics relate closely to object classes within the training set, such as cars, bicycles, faces, signs, trees, and windows. (In comparison with the recent results of Winn *et al.* [28], we should note that ours are obtained completely automatically from a large corpus of unlabeled images, whereas theirs are computed from a small set of single-object-category images.) These results show the power of classical segmentation methods augmented with the power of modern document analysis methods.

Acknowledgements: Financial support was provided by EC Project CLASS, NSF CAREER award IIS-0546547, the National Geospatial-Intelligence Agency, NEGI-1582-04-0004, and a grant from BAE Systems.

References

[1] T. Berg, A. Berg, J. Edwards, R. White, Y. W. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. CVPR*, pages 848–854, 2004.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proc. CVPR Workshop on Perceptual Organization*, 2004.

[4] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proc. CVPR*, 1997.

[5] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

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

[7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, 2005.

[8] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. ICCV*, Oct 2005.

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[10] A. W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *Proc. ECCV*, volume 3, pages 304–320. Springer-Verlag, 2002.

[11] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[12] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 43:177–196, 2001.

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

[14] http://www.robots.ox.ac.uk/∼vgg/research/affine/.

[15] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, Sep 1999.

Figure 5. Top segments for 4 topics (out of 10) discovered in the Caltech dataset. Note how the discovered segments, learned from a collection of unlabelled images, correspond to motorbikes, faces, and cars.

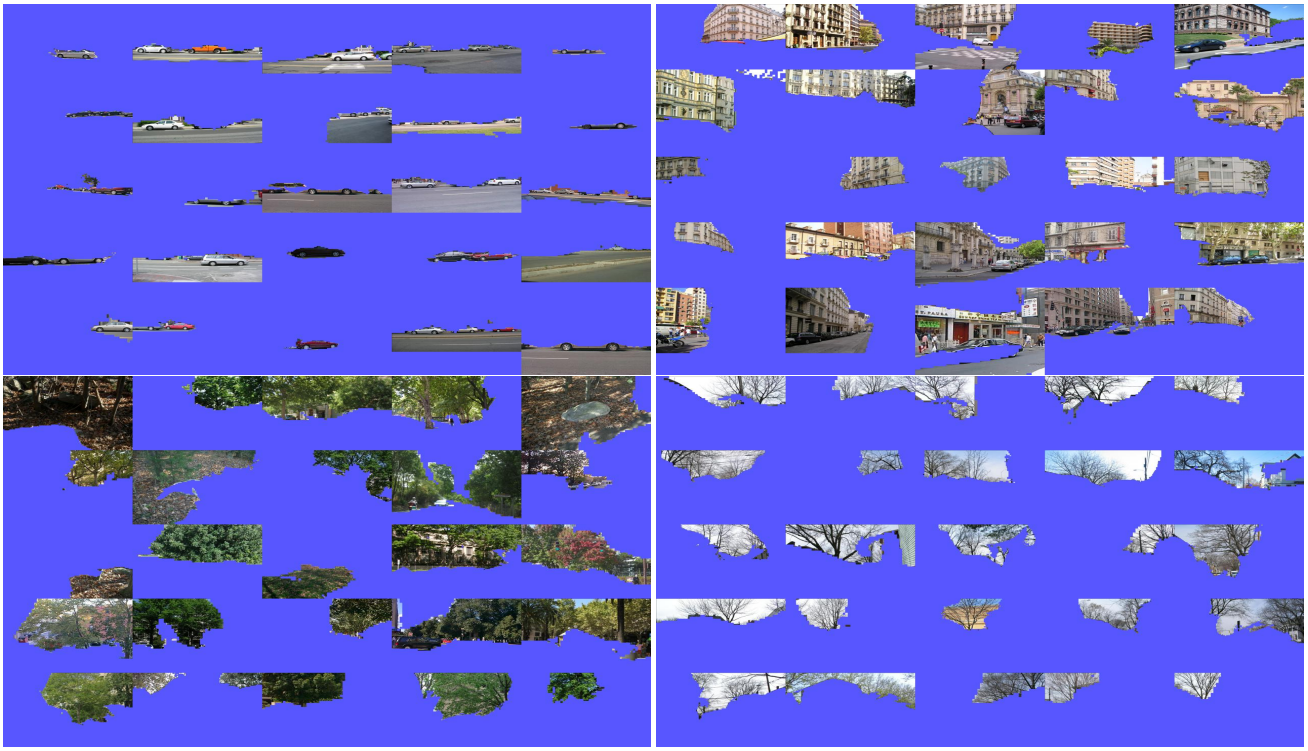


Figure 6. Top segments for 4 (out 20) topics discovered in the LabelMe dataset. Note how the discovered segments, learned from a collection of unlabeled images, correspond to cars, buildings, and two types of trees.

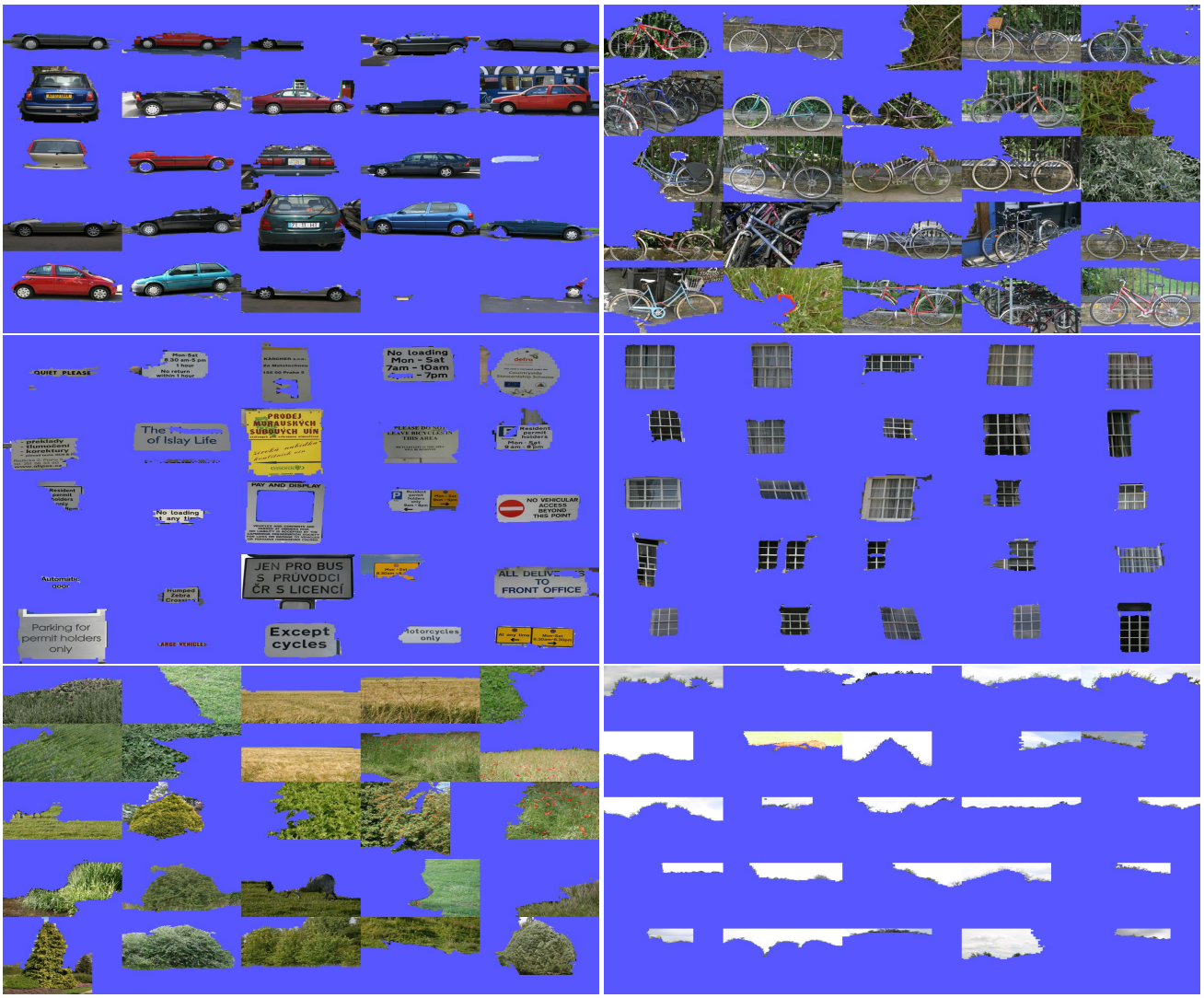


Figure 7. Top 21 segments for 6 topics (out of 25) discovered in the MSRC dataset. Note how the discovered segments, learned from a collection of unlabeled images, correspond to cars, bicycles, signs, windows, grass, and sky categories, respectively.

[16] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *NIPS*, 2002.

[17] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *Proc. ICCV*, 2005.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. Technical report, MIT AI Lab Memo AIM-2005-025, 2005.

[19] E. Sharon, A. Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. In *Proc. CVPR*, pages I:469–476, 2001.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. CVPR*, pages 731–743, 1997.

[21] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Proc. ICCV*, 2005.

[22] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, Oct 2003.

[23] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. CVPR*, 2004.

[24] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proc. ICCV*, 2005.

[25] L. Tolstoy. *Anna Karenina*. 1877.

[26] Z. W. Tu and S. C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE PAMI*, 24(5):657–673, 2002.

[27] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. ICCV*, 2005.

[28] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. ICCV*, 2005.