UNIVERSITY OF CALIFORNIA, SAN DIEGO

Image Segmentation and Contextual Modeling for Object Recognition

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Computer Science and Engineering

by

Andrew Rabinovich

Committee in charge:

Professor Serge Belongie, Chair
Professor Sanjoy Dasgupta
Professor David Kriegman
Professor Truong Nguyen
Professor Nuno Vasconcelos

2008

The dissertation of Andrew Rabinovich is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2008

DEDICATION

To my parents.

EPIGRAPH

No shot is a sure miss. – unknown

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

When I was a little boy, back in Soviet Union, my father used to invite his colleagues and students to our house, to work or just have dinner with. Being a son of a professor, who in turn was a son of a professor, I was always surrounded by conversations far too complex to grasp at that age. Yet, whenever a question arose, my parents and their friends made everything sound so simple and obvious. Somehow everything was logical and made perfect sense. Growing up in this environment gave me an early fascination with the academic world. It is my family that first made me consider an academic career. Starting at the age of 12, when my father asked me when I am going to get a Ph.D. from Harvard, and all the way until today, they have always given incredible support and helped with anything I needed. My wife, Irina, who joined this journey right at the beginning graduate school, took care of everything needed to allow me pursue my goals. Being a professor herself, she always understood the issues I dealt with, and helped in every possible way. I am eternally grateful to her.

With respect to my interest in computer vision, it all started at the Quantitative Microscopy Lab with Jeff Price, my undergraduate advisor. Hired as a programmer analyst to help automate microscopes, I soon became fascinated with image processing and automated cancer detection. Hacking simple scripts, I quickly realized how complex the field of image processing was. Without formal education, it seemed impossible to develop complex algorithms and achieve systematic results. I decided to take a class in image processing. Coincidentally, Serge Belongie just joined the department of Computer Science and was scheduled to teach this course. Since taking the course, Professor Belongie became my, at first unofficial, academic advisor. He always managed to get through my stubbornness and was always able to get the gist of my, at time poorly formulated, ideas. After working with Serge for over 8 years, he has not only become my mentor, but also a close friend. As part of the Belongie research group, $SO(3)$, I had the privilege of

working closely with some of its members. I would like to thank Sameer Agarwal, Kristin Branson, Vincent Rabaud, Boris Babenko, Carolina Galleguillos and Piotr Dollar for providing such a fruitful and comfortable working atmosphere. I have also had the opportunity to work with graduate students from ETH Zurich and UCLA. With Tilman Lange we introduced the notion of stable segmentations and with Andrea Vedaldi we proposed a new model of object recognition that includes segmentation and contextual reasoning.

Also, I would like to thank students and faculty of in the Computer Science department. In particular, I have have had many scientific and personal discussions with Sanjoy Dasgupta, David Kriegman, and Gary Cottrell. They have given me numerous insightful suggestions and have advised me in various aspects of computer vision, machine learning and general computer science. I also had the pleasure of working with students outside of my research group namely Lawrence Cayton, Eric Wiewiora, and Daniel Hsu. My gratitude extends to my thesis committee members as well; they have spent time with me and provided expert advice into my research.

Finally, being at the department of Computer Science at UCSD for almost eight years, I am mostly grateful to all the staff who have dealt with me and helped get past various hurdles and numerous complications.

VITA

1980	Born, Nizhny Novgorod, Russia.
2003	B.S., Computer Science and Engineering, University of California, San Diego.
2008	Ph.D., Computer Science and Engineering, University of California, San Diego.

PUBLICATIONS

"Weakly Supervised Object Localization with Stable Segmentations." Galleguillos C., Babenko B., Rabinovich A., and Belongie S., Proceedings of European Conference on Computer Vision, 2008, *to appear*.

"Object Categorization using Co-Occurrence, Location and Appearance." Galleguillos C., Rabinovich A. and Belongie S., Proceedings of Computer Vision and Pattern Recognition, 2008.

"Quantitative Spectral Decomposition for Stained Tissue Analysis." Rabinovich A., Agarwal S., Krajewska M., Krajewski S., Reed J., Price J.H., and Belongie S., IEEE Transactions on Medical Imaging, 2008, *in review*.

"Objects in Context." Rabinovich A., Vedaldi A., Galleguillos C., Wiewiora E. and Belongie S., Proceedings of International Conference on Computer Vision, 2007.

"Does Image Segmentation Improve Object Categorization?" Rabinovich A., Vedaldi A., and Belongie S., UCSD Technical Report cs2007-0908, 2007.

"Framework for Parsing, Visualizing and Scoring Tissue Microarray Images." Rabinovich A., Krajewski S., Krajewska M., Shabaik A., Hewitt S.M., Belongie S., Reed J.C., and Price J.H., IEEE Transactions on Information Technology in Biomedicine, 2006.

"Model Order Selection and Cue Combination for Image Segmentation." Rabinovich A., Lange T., Buhmann J., and Belongie S., Proceedings of Computer Vision and Pattern Recognition. 2006.

"Functional Proteomics of Cell Migration." Shen F., Hodgson L., Pertz O., Rabinovich A., Hahn K., Price J.H., Cytometry. 2006.

"Partitioning scale space." Rabinovich A. and Belongie S., UCSD Technical Report, 2005.

"Advances in Molecular Labeling, High Throughput Imaging and Machine Intelligence Portend Powerful Funcional Cellular Biochemistry Tools, Journal of Cellular Biochemistry" Price J.H., Hahn K., Hodgson L., Hunter E.A., Krajewski S., Morelock M., Murphy R.F., Rabinovich A., Reed J.C., Heynen S., 2003.

"Unsupervised Color Decomposition of Histologically Stained Tissue Samples." Rabinovich A., Agarwal S., Laris A., Price J. and Belongie S. Proceedings of Neural Information Processing Systems 2003.

"Unsupervised Color Decomposition of Histologically Stained Tissue Microarrays." Rabinovich A., Sagarwal S., Belongie S., Price J.H., , Archives of Pathology and Laboratory Medicine, 2003.

"Automated Analysis of Tissue Microarrays." Rabinovich A., Price J.H., Archives of Pathology and Laboratory Medicine, 2002.

"Scanning and Visualization of Tissue Microarrays." Rabinovich A., Price J.H., Journal of the Association for Laboratory Automation, 2002.

ABSTRACT OF THE DISSERTATION

Image Segmentation and Contextual Modeling for Object Recognition

by

Andrew Rabinovich

Doctor of Philosophy in Computer Science and Engineering

University of California, San Diego, 2008

Professor Serge Belongie, Chair

Recognizing objects is an essential part of navigating through the visual world. Identifying objects and finding boundaries between them provides us with some of the richest sensory information. Similarly, image segmentation and object recognition are among the most fundamental problems in computer vision and machine intelligence. The potential interaction between these processes has been discussed for many years. The usefulness of recognition for segmentation was demonstrated with various top-down segmentation algorithms; however, the impact of bottom-up image segmentation for object recognition is not well understood. One impeding factor is the unsatisfactory quality of image segmentation algorithms. In this work, we take advantage of a recently proposed method for computing multiple stable segmentations and illustrate the application of bottom-up image segmentation as a preprocessing step for object recognition.

In parallel to segmentation, the task of visual object recognition is often greatly facilitated by the objects' surroundings. Contextual information can play

the very important role of reducing ambiguity in objects' visual appearance. In this dissertation, we propose a new model for object recognition that incorporates two types of context – co-occurrence and relative location – with local appearance-based features, thus named CoLA (for Co-occurrence, Location and Appearance).

Since a number of contextual models for recognition have been proposed in the recent history, it is necessary to compare the newly proposed model to the existing ones. Over the years, two general kinds of such models have emerged: those with contextual inference based on the statistical summary of the scene, and models representing the context in terms of relationships among objects in the image. Understanding the theoretical and practical properties of such approaches is essential in designing object recognition systems. We provide an analytical analysis of these models and evaluate them empirically.

Chapter 1.

Introduction

Object recognition and categorization have been active topics of research in psychology and computer vision for decades. Initially, vision scientists and psychologists formulated hypotheses about models of object categorization and recognition, see [28, 29, 89]. Subsequently, in the past 10 years or so, object recognition and categorization have become very popular areas of research in computer vision. With two general models emerging, generative and discriminative, the newly developed algorithms aim to adhere to the original modeling constraints proposed by vision scientists. For example, the hypothesis put forth by Biederman et al. [7], suggests five classes of relations between an object and its setting that can characterize the organization of objects into real-world scenes. These are: (i) *interposition* (objects interrupt their background), (ii) *support* (objects tend to rest on surfaces), (iii) *probability* (objects tend to be found in some contexts but not others), (iv) *position* (given an object is probable in a scene, it often is found in some positions and not others), and (v) *familiar size* (objects have a limited set of size relations with other objects).

Classes (i, ii, iv, and v) have been addressed fairly well in the models proposed by the computer vision community, as shown in [11, 20, 87]. Class (iii), referring to the contextual interactions between objects in the scene, however, has

received comparatively little attention.

A large body of evidence in the literature on vision science, for example [8, 14, 22, 26, 34, 70], computer vision [23, 33, 76, 75, 92, 94] and cognitive neuroscience [1, 2, 3, 27, 67], has shown that contextual information affects the efficiency and accuracy of object recognition by humans and machines. There is a general consensus that objects appearing in a consistent or familiar background are detected more accurately and processed more quickly than objects appearing in variable scenes. Researchers in computer vision have recognized the importance of context and advocated its use for object recognition for many years [21, 88].

Existing context based methods for object recognition and classification consider global image features to be the source of context, thus trying to capture object class specific features. In [31, 60, 91, 102], the relationship between context and object properties is based on the correlation between the statistics of low-level features across the image that contains the object, or even the whole object category.

Semantic context[1] among objects has not been explicitly incorporated into existing object categorization models until very recentry. Semantic context requires access to the referential meaning of the object [7]. In other words, when performing the task of object categorization, objects' category labels must be assigned with respect to other objects in the scene, assuming there is more than one object present. To illustrate this further, consider the example in Figure 1.1. In the scene of a tennis match, four objects are detected and categorized: "Tennis court", "Person", "Tennis Racket", and "Lemon". Using a categorization system without a semantic context module, these labels would be final; however, in context, one of these labels is not satisfactory. Namely, the object labeled "Lemon", with an appearance very similar to a "Tennis Ball" is probably mis-labeled, due to the ambiguity in visual appearance. By enforcing semantic contextual constraints, provided by an oracle, the label of the yellow blob changes to "Tennis Ball", as

[1]We will use context and semantic context interchangeably from now on.

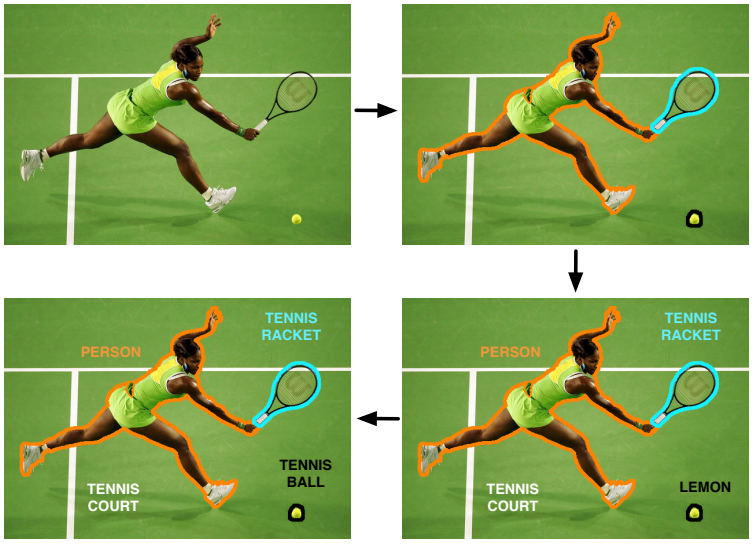


Figure 1.1: A possible idealized model for object recognition. An original image is segmented into objects; each object is categorized; and object labels are adjusted with respect to semantic context in the image. As a result, the label of the yellow blob changes from "Lemon" to "Tennis Ball".

this label fits in context with other labels more precisely. Note that, to achieve high recognition accuracy, this model heavily relies on *correct* segmentation of objects in images. Conventionally, it has been thought and argued that identifying object boundaries is rather impossible without top-down processing, i.e., must know the object identity prior to segmentation.

In this work we take advantage of a recently proposed method for computing multiple stable segmentations and illustrate the application of bottom-up image segmentation as a preprocessing step for object recognition and categorization. In Chapter 2, we extend a popular bag-of-features (BOF) recognition model to provide multiple class categorization and localization of objects in images. As post-processing, we propose to use contextual relations between objects' labels to help satisfy semantic constraints. With object categorization in hand, a conditional random field (CRF) formulation is used to maximize the objects' labels contextual agreement in Chapter 3. Chapter 4 provides an analytical and empirical

comparison of two main classes of contextual models for object recognition. We conclude with the discussion of the proposed methods and more general remarks about object recognition, image segmentation and contextual modeling in Chapter 5.

Chapter 2.

Image Segmentation for Object Recognition

The interplay between image segmentation and object recognition has been an active area of research for several decades, both in computer vision and cognitive psychology. The benefits of object recognition have been exploited in top-down image segmentation approaches. Combining object model knowledge and the initial low level segmentation has been shown to improve segmentation accuracy [10]. [48] introduced a principled way to improve top-down segmentations with low level features. However, the effects of image segmentation on object recognition and categorization are still not clear.

Discovering global structure is at the heart of most approaches to image segmentation. For example, image segmentation methods based on spectral clustering proceed by computing local measurements around each pixel followed by a partitioning step that aims to minimize a global cost function defined on pairwise affinities over these measurements [6, 62, 84]. In this setting, the global structure is represented concretely by a set of partition vectors indicating group membership. Many leading recognition engines, however, are solely based on local feature descriptors [13, 19]. Yet in contrast, the principle of global precedence suggests

that global image structure and configurations dominate local feature processing in human pattern perception and recognition [35, 61].

Recently, there have been efforts that leverage manually segmented foreground objects from the cluttered background to improve categorization. In [63], for example, flowers are segmented from the background to increase recognition accuracy. By segmenting the objects of interest, the noise introduced by the background around the object is minimized. Yet, methods of unsupervised image segmentation have not been popular as pre-processing for recognition and categorization. One reason for this, is the unsatisfactory quality of image segmentation algorithms. It is generally hard to find segmentations that capture all correct object boundaries in images of real world scenes. If the segments were satisfactory, an ideal segmentation based recognition system would resemble the sketch in Figure 2.1. After perfect segmentation, each segment (representing an object) is labeled by the recognition engine. Segment boundaries are used for localization and the scene category label is inferred from the individual object labels.

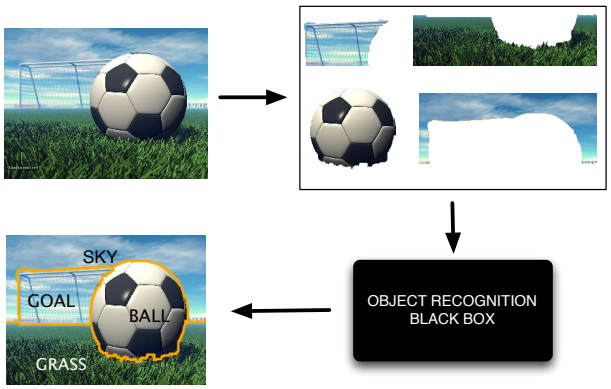


Figure 2.1: Illustration of a segmentation-based object recognition system. **Top Left:** Original image with four objects: soccer ball, goal, grass and sky. **Top Right:** Ideal image segments. **Bottom Right:** Discriminative object recognition system, e.g. "Bag of Features". **Bottom Left:** Multi-class object recognition with localization.

Existing recognition algorithms that advocate the use of segmentation ap-

pear to work well if strong initial object hypotheses are built into the segmentation engine [47, 103]. For the task of detecting and recognizing objects in still images without object knowledge, however, the recognition capability is still very weak, perhaps due to the segmentation performance. For example, the approach of Martin et al. [54] attempts to integrate all necessary visual cues together to produce one "best" segmentation. The work of Mori et al. [59] acknowledges that an erroneous segment boundary will degrade recognition accuracy, and thus proposes to oversegment an image into super-pixels to increase the potential quality of a particular merged segmentation. Alternatively, works such as [99] and recently by [42], suggest that attempts to calculate a segmentation for an input image are likely to introduce more harm than good, and that a bounding box, at every possible location and scale in the image, must be considered as an object outline for satisfactory object recognition and categorization performance. Thus, rather than partitioning an image into semantically meaningful parts, it may be possible to capture some global statistics of the image with bounding boxes of various scales. We will return to global image statistics and scene identification in Chapter 4.

A reason for the inadequate performance of image segmentation is the ambiguity of the image representation, the model parameterization, and the task itself. As described in [74], in general there does not exist a single correct segmentation of an image, but rather there exists a shortlist of meaningful image partitionings. Thus, unlike the above mentioned approaches of using a single segmentation or all possible bounding boxes, the idea of using several segmentations has recently emerged [23, 53, 74, 76, 78, 79]. A handful of segmentations is chosen in hope that a collection of all segments from these few segmentations will result in adequate object boundaries. Russell et al. [79] rely on a collection of hundreds of random segments to perform object detection, while we advocate the use of stability as a predictor of "goodness" of a particular set of parameters, cue weightings and model order, as done in [44, 74] to perform object recognition and categorization. Only the few most stable segmentations that depict various aspects of the image

are chosen to describe object boundaries. In this regard, the segmentations we use go beyond what is available via a simple oversegmentation or superpixel representation in terms of capturing salient image structure. The notion of stability, although defined quite differently, has been used in computer vision before. Matas et al. [55] introduced the maximally stable extremal region (MSER), which is a connected set of pixels obtained by intensity thresholding, the area of which is stable with respect to perturbations in the threshold value.

Partitioning images into segments has been proposed for learning the joint distribution of image regions and words for image region annotation [4]. Recently [78] suggested using multiple segmentations for object recognition. They build a segmentation based recognition system and report competitive results. However, they do not show the performance of their system without segmentation. Thus the effects of segmentation on object categorization remain unclear. Also they do not leverage segmentation for object localization and multi-class object recognition.

In this chapter we show that preprocessing a query image by representing it as a shortlist of segmentations increases the accuracy of object recognition. Having classified each of the segments we infer the following from the shortlist of segmentations: (a) a label for each segment, (b) object localization via the segment boundary, and finally (c) a label for the entire image. We evaluate the benefits of image segmentation, as pre-processing, for object categorization on the Caltech and PASCAL databases. In investigating the importance of image segmentation for object categorization by answer the following questions:

1. Can segmenting an image improve object recognition?

2. How does the number of segments affect recognition accuracy?

3. Does the quality of segmentation affect recognition accuracy?

4. Is it beneficial to perform localization using segmentation?

2.1 Stability based Clustering

Image segmentation is an instance of a clustering problem. In the domain of images, pixels that are similar, according to some criterion, should be clustered together; pixels that belong do different objects should be in different groups. Here we review the fundamentals of stability based clustering and its application to image segmentation.

The goal of clustering (or segmentation, or grouping) is to partition n objects into k groups so as to optimize an objective function. One way of thinking of the objective function is that it imposes a ranking on the set of all partitions. While this is a convenient tool for intuition, when k is unknown, the size of this set – the Bell number $B(n)$ – grows super-exponentially in n. For example, $B(100) \approx 4.8 \times 10^{115}$. Compounding the problem is the fact that most clustering algorithms possess a variety of parameters on the objective function that weight different features (or cues) of the objects. In the case of image segmentation, these features include position, color, texture, motion, and so on. As such, the problems of choosing k (model order selection) and the relative parameter weightings (cue combination) are difficult open problems.

Fortunately, the various domains in which clustering is applied often enjoy properties that can be leveraged against the above problems. In this work, our domain of interest is visual grouping. In this setting, k is often fairly small, e.g., 10 (i.e., representing the objects in the image), and the various parameters can be restricted into narrow valid ranges. Nonetheless, depending on the number of cues employed and the granularity of their variation, this can still present substantial problems both in the sense of computation and of usability.

This chapter addresses both of these problems. We begin with the observation that no single value of k is correct in general. The literature on model order selection is perhaps surprisingly focused on selecting one 'best' value of k [5, 15, 44, 49, 90]. A similar situation exists in the scale selection litera-

ture [38, 50]. We stop short of exhaustively searching the space of parameter values, however, by observing that this space has implicit structure, and that structure allows one to characterize the space via an efficient sampling scheme. The proposed clustering algorithm frees the user from the hassles of parameter tuning and model order selection: the input is a set of points, the output is a ranked shortlist of clusterings. The lingua franca we adopt in pursuit of this framework is a quantification of a *stable clustering*, corresponding to the intuition that a clustering is good if it is repeatable in the face of perturbations.

As a preview of this idea, consider the dataset shown in Figure 2.2. This

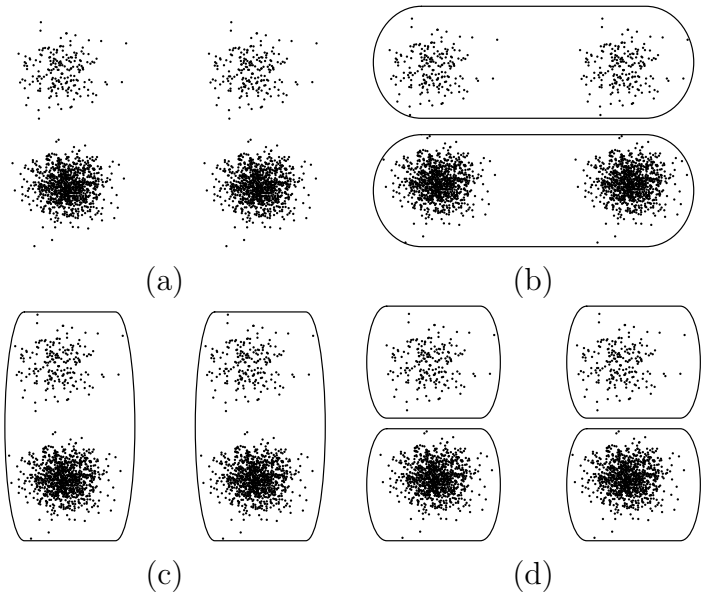


Figure 2.2: (a) Original stimulus of four clumps of points with varying density. Stable clusterings for $k = 2, 4$ are shown in shown in (b-d); (b) based on point density (simplest form on texture description), $k = 2$; (c) based on Euclidean distances between data points, $k = 2$; (d) based on Euclidean distances between data points, $k = 4$. There are two other trivial stable solutions for $k = 1$ and N, where N is the cardinality of the set. Note, no stable clustering exists for $k = 3$ with the given cues.

stimulus consists of four clumps of points, each drawn from a symmetric Gaussian distribution of different variance. For simplicity, we consider two cues: proximity and density, where the latter is measured by counting the number of points that

fall inside a box centered on each point.[1] We show three representative stable clusterings, two for $k = 2$ and one for $k = 4$. Depending on the relative cue weighting, one can obtain two very different stable clusterings for $k = 2$. For $k = 4$, however, there is only one stable clustering. Associated with each case is a range of parameter values (cue weighting and box width) that lead to the same result. The myriad other unstable segmentations are not of interest to us. Here we aim to select only meaningful clusterings for a given dataset. In particular we hope to select all stable clustering solutions.

Stability based clustering is a relatively new approach to model order selection. In late 1980s Jain and Dubes [36] discussed the validity of a given clustering structure based on hypothesis testing. The boom of work on finite mixture models in the 1990s gave rise to numerous approaches based on information theoretic criteria such as MDL, AIC, and BIC, see [9]. More recently a class of approaches based on stability have shown great promise. Our work falls into this category; we provide a brief review of it next.

2.1.1 Model Order Selection

The framework of stability based model order selection (from [44]) is as follows. Given a dataset, the data points are split into two disjoint subsets $\mathcal{A}$ and $\mathcal{B}$. Using some clustering method, cluster $\mathcal{A}$ into k groups. Once the clustering for a given k is done and all the points in $\mathcal{A}$ are labeled, a classifier ϕ is chosen and is trained using the labels from the clustering algorithm. Once the classifier (the predictor) is trained, the subset $\mathcal{B}$ is considered. The data in $\mathcal{B}$ is clustered into k groups and independently labeled using ϕ. Then the labels from the clustering and classification are compared to determine the stability. Care must be taken here since the labeling is arbitrary up to a permutation. To address this, one can perform pairwise comparison between points (i.e., are the two points in the

[1] This measure of local density is a simplified instance of a texture descriptor; it generalizes to a local texton histogram [52]

same group or not) or find an optimal label permutation, e.g., using the Hungarian method, [39]. Finally, the number of points with the same label provide a stability measure for that value of k. This procedure is repeated for a range of k's.

This approach is well motivated and we adopt it with the following modifications.

1. Firstly, instead of splitting the data into two subsets, cluster the entire data with a given k. The clustering engine can be a central method such as k-means if the clusters are spherical, or a pairwise method such as NCut if not; for generality assume that data is not spherical and always use a pairwise method. Note that NCut is a spectral clustering method, which may be considered as a pre-processing step for k-means. Also, NCut incorporates a compactness criterion, thus NCut can be reduced to k-means in an embedded space.

2. Once the data is clustered and the data points are labeled, add noise (proportional to the variance of the data (discussed in Section 2.1.3)) to slightly perturb the pairwise distances.[2] Once the data is perturbed, perform clustering for the same number of groups and assign new labels to the data. Such a labeling scheme avoids the use of a classifier, and reduces the algorithmic complexity. This perturbation is performed T times; here we re-clustered the data 50 times, yielding 50 different labelings for the data points.

3. Given all of the labellings, permute all but one of them to best match the hold-out set (anchor) and compute the stability according to the following definition:

$$\Phi(k) = \frac{1}{n - \frac{n}{k}} \left(\sum_{i=1}^{n} \sum_{j=1}^{T} \delta_{ij} - \frac{n}{k} \right). \tag{2.1}$$

Here n is the number of data points and δ_{ij} is equal to $\frac{1}{T}$ if the i-th point is mapped to the same cluster in the j-th perturbed grouping and zero otherwise. Fraction $\frac{n}{k}$ prevents from a bias to a particular value of k (this is an average cluster size for a given k); $n - \frac{n}{k}$ is the normalization coefficient. Thus Φ is a properly

[2] One could instead perturb the positions of each data point.

normalized[3] measure of the probability of a data point to change label due to a perturbation of the data set. Since any given anchor could be suboptimal, we try all possible anchors, and pick the one that yields highest stability. Clusterings with high stability scores are considered meaningful and are retained. Note that in general, there may exist several stable groupings.

2.1.2 Visual Cue Combination

Stability based model order selection is a method of choosing the appropriate number of groups, k, for the given clustering problem. Although this approach is native to learning theory, we apply it to computer vision and address the problem of cue combination (feature selection in machine learning).

Visual grouping is an instance of a clustering problem based on features such as color and texture. Similarly to the problem in clustering, the number of segments, or groups, is also unknown. We use stability based model order selection to determine the number of segments for a given segmentation instance. Unlike the case of point sets, where Euclidean distance may be used to assess the similarity between data points, image segmentation is best performed using multiple visual cues [52, 71, 95]. How to choose which cues to consider for a given segmentation problem and how to weigh their importance is unclear. This is known as the cue combination problem in computer vision. Traditionally, supervised learning approaches are used to address cue combination in a given application. Based on labeled data, a classifier is trained to choose the appropriate weighting for each cue, see [54, 56]. Since we do not assume human labeled examples are available, we propose to use the stability based approach to identify all possible combinations of cues and number of groups that lead to a stable segmentation.The stability calculation process remains unchanged; however, with every new combination of cues, the grouping criterion changes.

[3]In particular, $\Phi \in [0, 1]$ and it is not biased towards a particular value of k.

One possible approach to combining cues is to construct a similarity, or dissimilarity, matrix for each cue and combine them into a single affinity using a convex combination:

$$W_{ij} = \mathrm{e}^{-\sum_{f=1}^{F}(p_f \cdot C_{ij}^f)}, \text{ subject to } \sum_{f=1}^{F} p_f = 1, \tag{2.2}$$

where W_{ij} is the overall affinity between points i and j, C_{ij}^f is the similarity between the i-th and j-th point (pixel) according to some cue f, F is the number of cues, and $p_f \in [0, 1]$ specifies the cue weighting. Each of the similarity matrices C^f and affinity W have an internal scaling parameter, σ, that is used to maximally separate the dissimilar and group the similar entries in the matrix. We discuss the selection of this parameter next.

2.1.3 σ Estimation and Re-Sampling using Non-Parametric Density Estimates

In this section we estimate the scaling parameter for each cue individually and propose a re-sampling scheme for data perturbation. We assume that similarities C_{ij} correspond to *squared Euclidean* distances $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ in a suitable (embedding) space and that we have access to a vectorial representation of the data (one for each cue). If only similarities between data points are available, vectorial representation in $\mathbb{R}^d$, $d \leq n-1$ may be generated given that the similarity matrix fulfills the *Mercer's condition*, i.e., if we have a kernel matrix. Then, the application of Kernel Principal Component Analysis (kPCA) results in an *isometric* embedding of the corresponding distances into an $n-1$ dimensional space (after centering the kernel matrix).

σ Estimation. For each cue we obtain a set of n realizations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ of the random variable X with unknown density $p(\mathbf{x})$. The guiding principle of the stability approach is to require segmentations to be robust with respect to fluctuations in the source, i.e., in $p(\mathbf{x})$. If the density was known, this condition

would be easily checked for by drawing multiple samples from p. However, in practice, we do not have access to p. Thus, we adopt the following strategy. Instead of using the true (unknown) density p, we construct a non-parametric density estimate q, the latter being used for the re-sampling and the subsequent stability assessment.

A standard technique to obtain a non-parametric estimate of the density $p(\mathbf{x})$ is to use a *Parzen window* estimator: In essence, the density is approximated by a super-position of basis functions, the latter being centered around the realizations $\mathbf{x}_i$, $i \in \{1, \ldots, n\}$. One possible choice for the underlying kernel ($\simeq$ basis) function is a Gaussian kernel; the kernel centered at $\mathbf{x}_i$ reads:

$$k_{\mathbf{x}_i,\sigma}(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right). \tag{2.3}$$

The density estimate q_σ is then a super-positon of the individual density estimates:

$$p(\mathbf{x}) \approx q_\sigma(\mathbf{x}) := \frac{1}{n} \sum_{1 \le i \le n} k_{\mathbf{x}_i,\sigma}(\mathbf{x}). \tag{2.4}$$

The estimate depends on a smoothness (i.e., bandwidth) parameter, σ, whose choice greatly influences the shape of the density estimate q_σ. It is well known, that applying negative log-likelihood cross-validation (asymptotically) leads to a consistent estimate of σ, see [30] for more details. In essence, we wish to minimize:

$$\operatorname*{argmin}_{\sigma} \big(- \sum_{\mathbf{x}^{(t)} \in T} \log q_\sigma(\mathbf{x}^{(t)}) \big) \tag{2.5}$$

for different choices of σ and a set of "test" points T. Finally, a σ is picked, for which this test quantity becomes minimal.

Re-sampling. $k_{\mathbf{x}_i,\sigma}(\mathbf{x})$ is a Gaussian density with variance σ^2 and the corresponding density estimate q_σ is a mixture of Gaussians with n modes, each having weight $\frac{1}{n}$ and the common variance d-dimensional covariance matrix $\mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. This Gaussian mixture is used to get "noisy" versions of $\mathbf{x}_i$ by sampling from it. In particular, we get a noisy version $\tilde{\mathbf{x}}_i$ of each *original* point $\mathbf{x}_i$ by sampling a substitute

from the Gaussian $k_{\mathbf{x}_i,\sigma}$. Note that, in contrast to the original stability approach by Lange et al., the point correspondence problem is automatically resolved in this case, as one can identify $\tilde{\mathbf{x}}_i$ sampled from $k_{\mathbf{x}_i,\sigma}$ with $\mathbf{x}_i$.

Just Noticeable Difference. Besides the scaling parameter, some cues have another internal parameter, ω. For example the density cue has a window in which the density is computed. The size of such a window is an internal parameter of the density cue (other cues such as proximity do not have such internal parameters and simply use some properties of points to determine similarity). Varying the values of such internal parameters has a particular effect on the overall stability of the clustering. If for example the window size ω of the descriptive texture element is changed by a small fraction to $\omega + \epsilon$, the stability of a given grouping will not change, as texture is captured equally with windows of similar sizes. This is related to the phenomenon of the *Just Noticeable Difference* and *Weber's Law* [69]. Slight variations of the texture element window do not result in perceptually distinctive textures [32, 37, 43, 72]. Unfortunately, such a rule does not apply to the variation of number of groups. If the stability of segmentations with all possible cue combinations is known for k groups, in general there can be nothing said about the stability behavior of grouping with $k \pm 1$ groups. In the example in Figure 2.2 there are stable solutions for $k = 2$ and $k = 4$ groups, however, no clustering with $k = 3$ is stable.

2.1.4 All Possible Clustering Solutions for a Set of Cues

As discussed earlier, there may be more than one stable clustering for a given data set. Points may be grouped using different cue combinations and/or model orders. To identify all parameter settings for which a clustering is stable, it is intuitive to consider all such parameter values: different numbers of groups, different contributions of each cue, and finally the internal parameters for each of the cues. For example, again consider the point set in Figure 2.2. We would like

to find all stable solution based on two cues: pairwise proximity and point density within a window. Even if we restrict the range of parameters, e.g. 10 different values for k, 20 values for window size, and 10 for cue combination, there are still 2,000 clusterings to consider, as shown in Figure 2.3. Although such a representa-

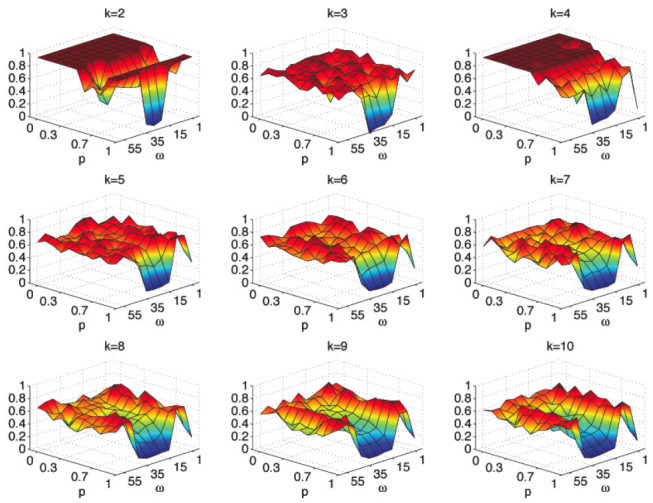


Figure 2.3: Slices of the cube of all possible segmentations for the 4 clumps stimulus (shown in Figure 2.2); the number of groups is indicated on top of every slice. All the stability values are in the range of $[0, 1]$. p is the cue combination axis in $[0, 1]$, ω is the window radius, an internal texture parameter, $\in [1, 55]$. As expected, there are stable solutions for a range of cue parameters when grouping into $k = 2$ and $k = 4$ groups. It is important to note that although the slices for $k = 2$ and $k = 4$ show high stability, the slice for $k = 3$ is unstable. This underlines the decoupled behavior of the order in model selection.

tion of the space of possible solutions is very thorough and potentially useful, the brute force computation of these groupings and stability values associated with them becomes computationally inefficient. Instead, we propose a sampling based approach for approximating the space of clustering stabilities.

Since the behaviors of solutions for different k's are decoupled, we must sample a set of parameter values of cue combination for every desired k. Once the sample stability values are computed, in the space of all possible parameters (cube of all clusterings), we construct a dense matrix A by interpolating the sampled values for each k.

To be able to analyze the behavior of the parameters independently, as was discussed in Section 2.1.3, the overall stability of a given solution is modeled as a product of stabilities of each individual cue. With A, a matrix of clustering stabilities constructed, we use Non-negative Matrix Factorization (NMF) to decompose the overall clustering stability values into clustering stabilities of individual cue parameters. NMF, see [46, 68], is a recently introduced method for finding non negative basis functions (vectors) that represent the data. Using an iterative approach with non-negativity constaints, a data mixture A is factored into constituent components S and the weights B for each component. Repeated iteration of the update rules is guaranteed to converge to only a locally optimal matrix factorization, however, practical applications of NMF indicate suitability of the approach. In its usual form, this decomposition is an additive one in terms of the learned components. Here, we set up the problem as multiplicative and consider B and S to be the two basis functions. In doing the NMF, there is a constraint on non-negativity, yet there is no upper bound on the individual entries of the basis functions. Since the basis functions that we extract correspond to the stability value for individual cues for a given k, the entries in vectors B an S must be constrained between $[0, 1]$. To enforce the bounds on the values, we introduce an extension to the general NMF. In the presence of more than two cues, the use of NMF is generalized to Non-negative Tensor Factorization [83, 101]

2.1.5 Bounded Non-negative Matrix Factorizations

To achieve the desired bounds on the elements of B and S, the following procedure is based on a *rank*-1 decomposition; however, it is possible to achieve *rank* K decompositions of A with K *rank*-1 consecutive decompositions.

Given $A = BS$, subject to $0 \leq B_{ij} \leq 1$, $0 \leq S_{ij} \leq 1$, the decomposition is an outer product of B and S assuming $rank(A) = 1$. To constrain the upper bound of elements of B and S, we wish to re-write $A = BS$ in terms of a function

that is restricted on the interval $[0, 1]$. For example e^{-X}, if $X \geq 0$, is constrained in $[0, 1]$. Let $A' = -\log(A)$, $B' = -\log(B)$, $S' = -\log(S)$, thus $A' = B' + S'$ is an instance of a least squares problem.[4] In particular, let $V = A'(:)$ (concatenated), $b = [B'(:, 1); S'(1, :)]$, and let

$$X = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ \multicolumn{6}{c}{...} \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

X is $mn \times (m+n)$, where m and n are the lengths of B and S. Thus, $A' = B' + S'$ becomes:

$$V = Xb \tag{2.6}$$

In order to satisfy initial constraint of $0 \leq B_{ij} \leq 1$, $0 \leq S_{ij} \leq 1$, this least squares problem must be solved with the constraint that $b_i \geq 0$ (`lsqnonneg` in Matlab). By performing the above substitutions in reverse, B' and S' are recovered. Finally, we exponentiate $A' = B' + S'$ to obtain the bounded decomposition into B and S.

$$e^{-A'} = e^{-B'(:,1)} e^{-S'(1,:)} \tag{2.7}$$

$$A = BS \tag{2.8}$$

This is Bounded Non-negative Matrix Factorization with *Rank*-1 assumption.

Projection onto the subspace of the approximation. *Rank*-1 approximation of the stability matrix may not be sufficient to represent the structure of A accurately; a higher order approximation may be required. Thus, it is necessary to quantify the performance of such an approximation.

[4] Since A' is $m \times n$ and both B' and S' are vectors, the entries of B' and S' are repeated to achieve the $m \times n$ dimensions.

The subspace spanned by the original full *rank* matrix A is projected onto the subspace spanned by the eigenvectors of the approximated matrix Q. In the case of *rank*-1 approximation, A is projected onto a line – Q's only eigenvector. More formally, we would like to find a combination of $\sum x_i q_i = a_j$, where each a_j, a column of A, is projected onto the subspace spanned by eigenvectors of Q. Let the set of eigenvectors spanning Q's subspace be Q_e. In matrix form[5]:

$$Q_e X = A \tag{2.9}$$

$$Q_e^\top (A - Q_e X) = 0 \tag{2.10}$$

$$Q_e^\top Q_e X = Q_e^\top A \tag{2.11}$$

$$X = (Q_e^\top Q_e)^{-1} Q_e^\top A \tag{2.12}$$

X is the projection matrix of A onto the eigenspace of Q. Thus, the projection of A onto Q's eigenspace is $P = QX$ and the residual of the projection P and original matrix A is $E = A - P$. Figure 2.5 illustrates the *rank*-1 and *rank*-2 approximations of the original matrix A and the residuals between A and the approximations. With a *rank*-1 approximation the residual error is 31.7%, while the residual error of *rank*-2 approximation is only 4.26%.

Cue interpolation and overall approximation accuracy. Unlike k, a cue parameter such as point density within a window is not independent of its "neighbors". In particular, visual cues may have a piecewise constant or monotonically changing behaviors and may be modeled as such. By having only a few stability values along the discrete interval of values for a given cue, it is possible to use a simple model to interpolate to the rest of the desired values for a particular cue. Since the actual cue combination is modeled as a convex combination, the behavior of that "axis" is continuous and smooth, and was fit using a simple bicubic interpolation. The behavior of stability as a function of box size for point density

[5] $Q_e^\top (A - Q_e X) = 0$ comes from the diagram (the dotted line goes from a_j to the nearest point Qx_i in the subspace as shown in Figure 2.4. This error vector $(a_j - Qx_i)$ is perpendicular to the subspace and it makes a right triangle with all the vectors $q_{e1} ... q_{en}$; $Q_e^\top Q_e$ is invertible since the q_e's are orthogonal.

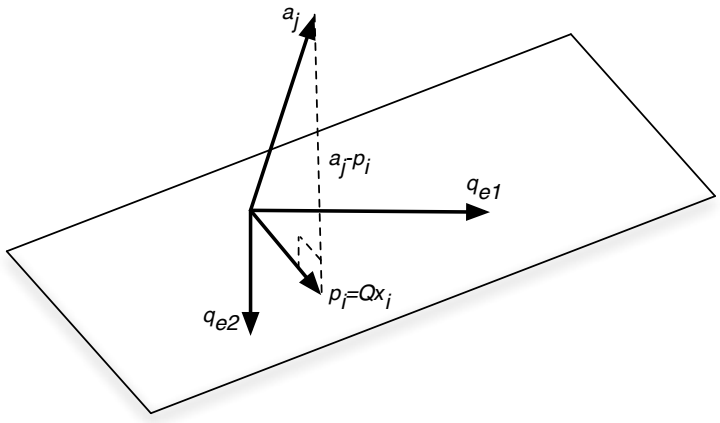


Figure 2.4: Rank-k error estimation. Columns of the original matrix A are projected onto the subspace spanned by the eigenvectors of the approximated matrix Q.

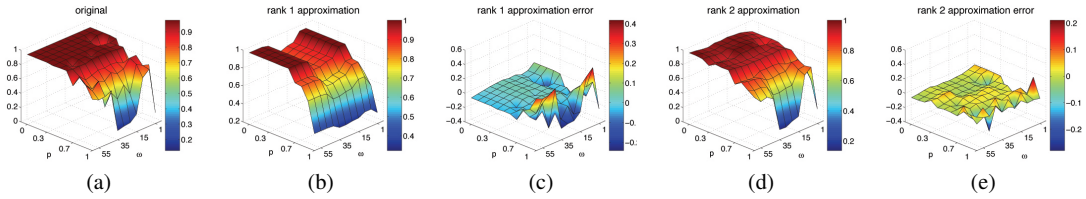


Figure 2.5: Accuracy of bNMF approximation of the stability matrix A from Figure 1 for $k = 4$. (a) Original; (b) *Rank*-1 approximation of A using bNMF; (c) Error of *rank*-1 approximation; (d) Two successive *rank*-1 approximations; (e) Error of *rank*-2 approximation.

was modeled as piecewise constant. Once the vectors B and S are filled, their outer product will fill the entire space of stability values for all segmentations for a given number of groups. To measure the quality of such a sampled approximation, we compare the approximated cube of segmentation to the actual one, where each stability value is computed according to our definition of stability. The approximation evaluation was carried out using the stimulus in Figure 2.2. As shown in Figure 2.6, with less than 20% of all possible combination of parameters for each k, an approximation of 90% accuracy is achieved. Accuracy was calculated via a

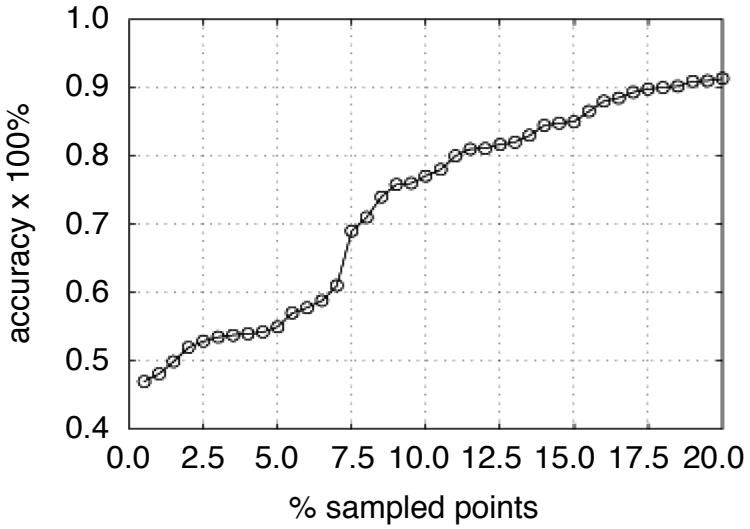


Figure 2.6: Evaluation of the accuracy of sampling the cube of stabilities to approximate its dense representation. The curve illustrates the agreement between two dense cubes of stability values, where the entries in the rst cube are all explicitly computed and the entries in the second are the result of sampling and interpolating. By sampling less than 20% (out of 200 points in each plane of the cube) of the full cube, the sampling approach is able to achieve an accuracy of 90%.

projection procedure in Section 2.1.5.

2.2 Shortlist of Stable Segmentations

Having introduced the stability heuristic for unsupervised learning, we now describe how to define and compute stable image segmentations. For each choice of cue combination $\vec{p}$ and number of segments k one obtains different segmentations of the image. Of all possible segmentations arising in this way, one or more can be considered "meaningful." For a choice of the parameters $\vec{p}$ and k, the image is segmented using Normalized Cuts [52, 84] using the implementation of [6]. Images are segmented using brightness and texture cues ($F = 2$). The segmentation is considered stable if small perturbations of the image do not yield substantial

changes in the segmentation. Segmentations with high stability score are considered meaningful and are retained. The stability heuristic may be used to select both the cue combination parameter $\vec{p}$ and the model order k. For the task of object recognition, we use stability to only resolve the cue combination problem. 11 values from $\vec{p}$ are chosen uniformly to compute the corresponding stability scores. The model order corresponds to the number of identifiable objects in the scene, and is sampled between $1 < k < 11$. In particular, the most stable segmentation for each value of k is demanded. In a sense, k not only represents the model order, but also the scale, as the model order increases, the individual segment size usually decreases.

In general, however, the set of stability values provides valuable information about the associated segmentations, but such a representation still requires manual sifting of the stable segmentations from the ones with low stability. Our aim is to output only a small set of parameter values that lead to stable segmentations. To identify stable solutions, we adopt a hysteresis based thresholding approach. Due to the continuous nature of the behavior of visual cues, we only consider regions of high stability values rather than individual points of high stability in this space of segmentations, to avoid noise. We begin pruning the cube by choosing a point of high stability with an assumption that every image has at least one stable segmentation; the stability values in the neighborhood are grouped into plateaux (each plateau represents a unique segmentation) by region growing, see [25]. We enforce that at least 2 neighboring positions have a high value to consider this region to be stable and result in a plateau. Sequentially, another point of high stability, outside of the explored plateau, is considered, and the region exploration is repeated. Such a process is repeated until all values of high stability (above a certain threshold) have been considered. Currently we set the upper, τ_u , and lower, τ_l , thresholds manually. In the ropes stimulus in Figure 2.7, for example, we set $\tau_u = 0.974$ and $\tau_l = 0.691$. Finally a set of parameters for cue combination, texture window size, and model order are output as a shortlist. This is the list of

all possible parameters that provide stable segmentations. With images presented in Figure 2.7, the shortlist of all possible stable segmentations reduced the size of the entire space of possible parameter combinations by more than 95%. The shortlist is a highly compact summary of the entire space of all segmentations. Some combinations of parameters may result in redundant segmentations and some segmentations may be stable but meaningless. Removing the parameters that yield incorrect and redundant segmentation is a subject of ongoing and future work.

There are a number of domains where the existence of multiple segmentations for a given image is natural; biomedical imaging is one of them. Due to the hierarchical nature of biological structures, segmentations with various numbers of groups are natural. Also, it is desirable to be able to identify segments based on different features (cues), e.g., DNA content, protein expression and brain activity. Here we present examples of multiple stable segmentations of images of tissue biopsy samples. To explore the generality of our framework, we apply it to images from the Berkeley Segmentation Database (BSD) as well. In Figure 2.7 are segmentations of three images from BSD and three images of tissue samples. In all six examples the different segmentations are the results of varying the number of groups and the cue weightings (using texture and color). Averaged segment boundaries, in the 4th column of the rst 3 rows, from multiple subjects from BSD (darker boundaries indicate higher probability for a given set of human segmentations) further illustrate the presence of multiple stable segmentations and exhibit a high correlation with segmentations produced by our method. Table 2.2 shows the segmentation statistics of our method for the images from Figure 2.7.

Similarity of texture was measured using texton histograms with an internal parameter of texture window radius [16]. Similarity of color was based on the Euclidean distance of the hue channel in HSV color space. Binning kernel density estimates of the color distribution in CIELAB color space using a Gaussian kernel, and comparing histograms with the χ^2 difference may be perceptually more meaningful; however, the choice of color description is not central here. We

Table 2.1: Multiple stable segmentations statistics. The percent of stable solutions is computed assuming that each of the plateaux of stable parameter combinations represents only 1 segmentation. In reality there could be more than 1, however, even if there are a few different segmentations per plateau, the fraction of stable ones will still be less than 5%.

stimulus	max k	total possible param. comb.	total # stable plateaux	% stable segmentations out of all possible	mean # of parameter combinations per plateau
ropes	20	5500	19	0.31%	29.4
clouds	20	5500	14	0.25%	45.7
flowers	50	13750	86	0.62%	8.1
tissue 1	100	27500	218	0.79%	11.6
tissue 2	100	27500	109	0.39%	38.9
tissue 3	100	27500	236	0.86%	12.2

chose the HSV representation for its simplicity [96]. Given the similarities for each cue, the overall pairwise pixel affinity was computed according to Equation 2.2. Once the combined affinity matrix W is constructed, using the proposed $rank$-1 sampling approach twice, its entries are treated as edge weights of an undirected graph. A number of graph based approaches cut such a graph based on some criterion [62, 84, 100]. In this work, we use Normalized Cuts implementation of [6], where the number of leading eigenvectors were set to k and were further thresholded using k-means clustering. The current algorithm is implemented in Matlab and on average takes 2.51 seconds of processing for each stability value on a dual 3.2 GHz processor with 2 GB of RAM. For example, the ropes image is 256×255 pixels and the full process took 0.76 hours (46 minutes) (without sampling it takes 3.83 hours). Note that current algorithm is highly parallelizable; this is exploited in Chapters 3 and 4.

In this chapter we proposed a framework that frees the user from the burden of manual parameter tuning and model order selection in the task of image segmentation. We leverage the observation that the number of possible segmentations of a dataset is signicantly smaller than the number of parameter combinations of the

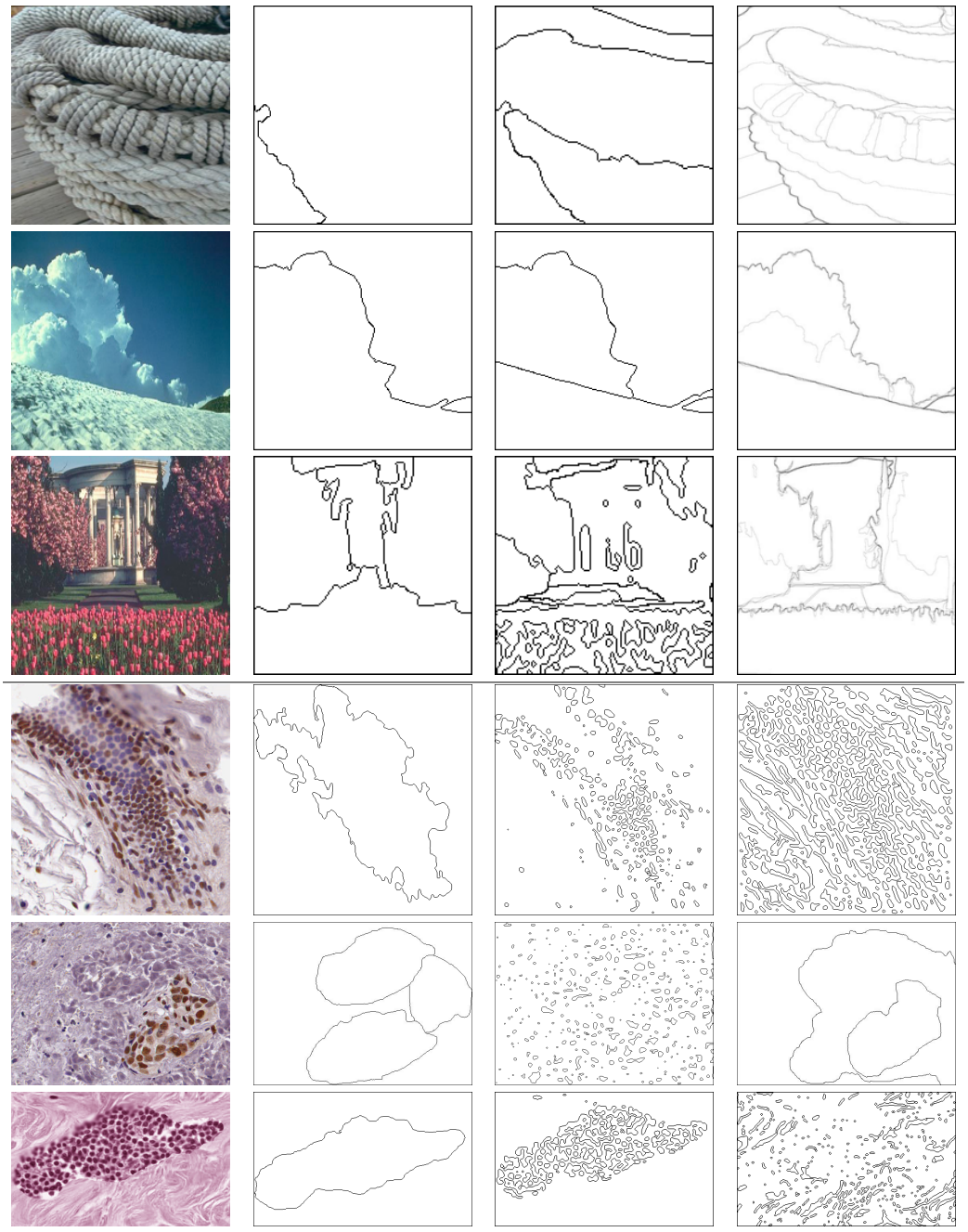


Figure 2.7: Examples of stable segmentations. Each is a result of a different cue combination and model order. Only two and three stable solutions are shown for the BSD and tissue examples, respectively. In all examples, over 95% of all possible segmentations have low stability and are discarded. In column 4 of the first 3 rows, we show averaged segment boundaries from multiple human subjects from the BSD (darker boundaries indicate higher probability for a given set of human segmentations).

segmentation algorithm; furthermore the number of stable segmentation is much smaller than the total number of possible segmentations. With an image as input, our method generates a shortlist of stable segmentations. The proposed approach performs well on medical images and examples from the Berkeley Segmentation Database. After sampling a small fraction of the possible cue weightings and internal parameters, the sparsely populated space of segmentation stabilities is filled in using a novel extension to NMF that constrains both the upper and lower bounds of the elements of the extracted basis functions. Stabilities for individual cues and the weights are interpolated to the desired resolution and the full space of segmentation stabilities is reconstructed. Finally, only parameters (k, cue weights and cue internal parameters) that result in stable segmentations are returned. The selected segmentations are stable, but not all may be unique. In future work we will address the problem of only including highly stable segmentations that are not redundant.

2.3 Integrating Bag of Features and Segmentation

With stable segmentations at hand, we now describe how image segmentation may be used as a pre-processing step for object recognition. In this work we utilize the Bag of Features (BoF) object recognition framework [13, 65] due to its popularity and simplicity. This method consists of four steps: (i) images are decomposed into a collection of "features" (image patches); (ii) features are mapped to a finite vocabulary of "visual words" based on their appearance; (iii) a statistic, or *signature*, of such visual words is computed; (iv) the signatures are fed into a classifier for labeling. All four steps can be implemented in a variety of ways. Here we adopt the implementation and default parameter settings provided by [97].

While a more sophisticated version of bags-of-features is likely to improve the categorization accuracy, we prefer to work with a simple implementation to better emphasize the effect of segmentation on the performance of the algorithm.We integrate segmentation with BoF as follows. Each segment is regarded as a standalone image by masking and zero padding the original image. Then the signature of the segment is computed as in regular BoF, but any features that fall entirely outside its boundary are discarded. Eventually, the image is represented by the ensemble of the signatures of its segments.

This simple idea has a number of effects: (i) by clustering features into segments we incorporate coarse spatial information; (ii) masking often enhances the contrast of segment boundaries, making features along the boundaries more shape-informative; (iii) computing signatures on homogeneous segments improves their signal-to-noise ratio.

Next we discuss how segments and their signatures are used to classify segments and whole images and to localize objects in them.

Labeling Segments. Let I be a test image and S_q its q-th segment, with i being the image index and c the category index, such that I_{ic} is the i-th training image of the c-th category. Let $\phi(I)$ (or $\phi(S)$) be the signature of image I (or segment S) and $\Omega(I)$ (or $\Omega(S)$) the number of features extracted from an image I (or segment S). Each image is partitioned into 9 different stable segmentations ($k = 2 \dots 10$), resulting in a soup of $2 + 3 + \cdots + 10 = 54$ different segments. Methods of [79, 53] also construct a so-called "soup" of segmentations, but those are random segmentations and typically thousands of segments are needed.

Segments are classified based on a simple nearest neighbor rule. Define the un-normalized distance of the test segment S_q to class c as:

$$d(S_q, c) = \min_i d(S_q, I_{ic}) = \min_i \|\phi(S_q) - \phi(I_{ic})\|_1 \tag{2.13}$$

So $d(S_q, c)$ is the minimum l_1 distance of the test segment S_q to all the training

images I_{ic} of category c. We assign the segment S_q to its closest category $c_1(S_q)$:

$$c_1(S_q) = \underset{c}{\operatorname{argmin}}\, d(S_q, c). \tag{2.14}$$

In order to combine segment labels into a unique image label it is useful to *rank* segments by classification reliability. To this end we introduce the following confidence measure.

Labeling Confidence. Define the *second best labeling* of segment S_q as the quantity:

$$c_2(S_q) = \underset{c \neq c_1(S_q)}{\operatorname{argmin}}\, d(S_q, c). \tag{2.15}$$

In order to characterize the ambiguity of the labeling $c_1(S_q)$ we compare the distance of S_q to $c_1(S_q)$ and $c_2(S_q)$, defining:

$$p(c_1(S_q)|S_q) = (1-\gamma) + \gamma/C, \quad \text{where } \gamma = \frac{d\left(S_q, c_1(S_q)\right)}{d\left(S_q, c_2(S_q)\right)} \tag{2.16}$$

and C is the number of categories. This is the belief that S_q has class $c_1(S_q)$; for other labels, $c \neq c_1(S_q)$:

$$p(c|S_q) = \frac{1 - p\left(c_1(S_q)|S_q\right)}{C-1}. \tag{2.17}$$

So $p(c|S_q)$ is a probability distribution over labels and it is uniform when $d\left(S_q, c_1(S_q)\right) \approx d\left(S_q, c_2(S_q)\right)$ and peaked at $c_1(S_q)$ when $d\left(S_q, c_1(S_q)\right) \ll d\left(S_q, c_2(S_q)\right)$. When $p(c|S_q)$ is indeed uniform, segment S_q is discarded to avoid a forced choice label. Empirically, we assume $p(c|S_q)$ is uniform iff $c_1(S_q)$ is within 10% $c_2(S_q)$.

Labeling Whole Images. Let $\{S_1, ..., S_K\}$ be all the segments of a test image I. We let the segments vote for the image label as follows. Each segment S_q votes for class c proportionally to its confidence $p(c|S_q)$ and has an amount of votes $w(S_q)$ to use. The label of the image I is then given by:

$$c(I) = \underset{c}{\operatorname{argmax}} \sum_{q=1}^{K} p(c|S_q) w(S_q). \tag{2.18}$$

The weights $w(S_q)$ encode both the importance and the reliability of the segment S_q, irrespective of the class label. As both of these factors are roughly proportional to the number of features of the segment, we define $w(S_q) = \Omega(S_q)/\Omega(S_{\max})$ where $S_{\max}$ is the largest segment (in terms of number of features).

Localization. In many approaches to object localization, the bounding box that yields highest recognition accuracy is used to describe objects' location [57, 99]. Here we use the segment boundaries instead.

Given the labels of each segment, $c_1(S_q)$, and the overall image label, $c(I)$, we look for segments whose labels match the image label, i.e., $c(I) = c_1(S_q)$. Among these, we check for overlapping segments and we return the first k unique segment boundaries. Note that this method is not limited to BoF and could be used to add localization capabilities to other recognition methods.

To recognize and localize objects of classes other than the image class, all segments S_q are ranked with respect to their label confidence $p(c_1(S_q)|S_q)$ and the first k segment boundaries are returned irrespective of the whole image label.

2.4 Effects of Image Segmentation on Object Recogntion

To answer the above formulated questions empirically, we performed categorization experiments on images from the standard datasets Caltech-101 and PASCAL. For the Caltech-101 database we picked the twenty most difficult categories, as reported by [104]. For both databases, we used 30 images per category for training. The implementation details of [97] for the BoF model are the following. 5000 random patches at multiple scales (from 12 pixels to the image size) are extracted from each image such that larger patches are sampled less frequently (as these would be redundant). The feature appearance is represented by SIFT descriptors [51] and the visual words are obtaining by quantizing the feature space

using hierarchical K-means with $K = 10$ at three levels [64]. The image signature is an histogram of such hierarchical visual words, L_1 normalized and TFxIDF re-weighed [64]. In an unoptimized MATLAB/C implementation, the computation of SIFT and the relevant signatures, takes on average 1 second for each segment in the image on on a Pentium 3.2 GHz. Finally, the signatures are fed to a k-nearest-neighbor classification algorithm. Implemented in MATLAB, training the classifier and constructing the vocabulary takes under 1 hour for 20 categories with 30 training images in each category. Classification of test images, however, is done in just a few seconds.

To understand the importance of image segmentation quality for object categorization accuracy we consider the following two segmentation methods. The first is the stability based segmentation described earlier. Implemented in MATLAB, each segmentation takes between 10-20 seconds per image with $T = 100$ restarts, on a Pentium 3.2 GHz , depending on the image size. Typical images in the Caltech database are at least 600×400 pixels. We'll refer to this method as "Stable Segmentations" (Sseg). The second segmentation method is a simple grid-like image partitioning method, similar to that of [45]. In real time, an image is broken into $k = 4, 9, 16, 25$ equal sub-images, which together results in 54 segments $(4 + 9 + 16 + 25)$. We refer to this method as "Block Segmentations" (Bseg).

2.4.1 Average Recognition Accuracy

We compare the categorization results of the BoF with and without segmentation pre-processing to quantify the effects of image segmentation on the accuracy of object categorization. Figure 3.2 shows the confusion matrices of 20 most difficult categories from the Caltech-101 and PASCAL databases simply using the BoF model. Confusion matrices of average recognition with no pre-processing, with "Block Segmentations", and with "Stable Segmentations" are shown in columns (a), (b), and (c) respectively. The results of average recognition accuracy are sum-

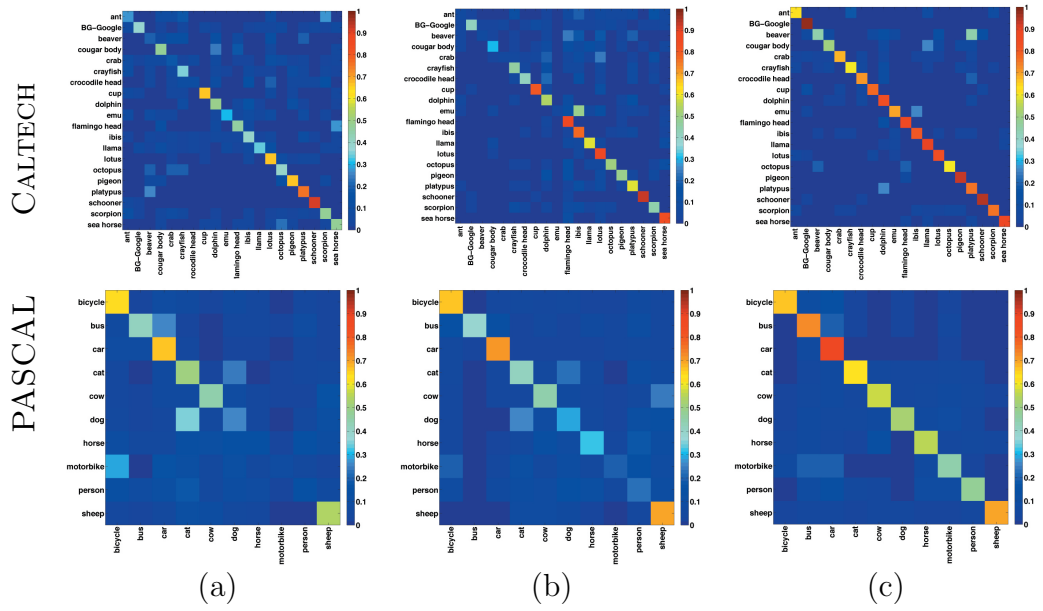


Figure 2.8: Confusion matrices of object categorization accuracy using the BoF model. **Top row:** 20 hardest categories of Caltech101. **Bottom row:** PASCAL dataset. (a) BoF model with no preprocessing. (b) BoF model with test images represented by "Block Segmentations". (c) BoF recognition model with test images represented by "Stable Segmentations".

marized in Table 3.2.2. The reported results are based on 54 segments per image. In the case of "Stable Segmentations" segments are taken from 9 segmentations, and for "Block Segmentations" from 4.

Table 2.2: Average object categorization accuracy for both the Caltech and PASCAL datasets. **No Seg:** Bag-of-Features model applied to the whole image. **Bseg:** Bag-of-Features model applied to the individual block segments of the image. **Sseg:** Bag-of-Features model applied to the individual stable segments of the whole image.

	No Seg	Bseg	Sseg
Caltech	44.9%	50.6%	75.5%
PASCAL	38.5%	43.5%	61.8%

2.4.2 Localization

The quality of object localization, whether for single or multi-class recognition, can be evaluated in a number of ways. Some compare object centroid location, while others attempt to maximize the overlap between predicted bounding box around the object and the ground truth one [57]. However, objects are generally not rectangular and should be localized by their boundary contour, which we do here. To quantify the accuracy of object localization, we adopt a method from the PASCAL Challenge [16] and consider the overlap, ρ, between ground truth localization, GT, and the retrieved localization, R, is $\rho = \frac{GT \cap R}{GT \cup R}$. Note that ρ is misleading in cases where the objects' contour area is smaller than that of its bounding box (Figure 2.11). In Table 2.3 we report the average localization accuracy for each category in both the Caltech and PASCAL datasets. For each image, the segment R, which is more likely to have a given label, is compared to the ground truth bounding box GT.

We have also explored the relationship between number of segmentations per image and object localization accuracy. Generally, categories of objects with complex boundaries are localized more accurately as the number of segments increase, while blob-like objects do not benefit as much from an increase in the number of segments. Figures 2.10 and 2.11 show examples of objects localized by our method.

2.4.3 Quality of Image Segmentation

Due to the principle of global precedence and the importance of the shape cue, it is expected that the object categorization accuracy based on "Stable Segmentation" should outperform that of the trivial "Block Segmentations". Indeed, the results in Table 3.2.2 indicate that the improvement with "Stable Segmentations" is significant.

The localization based on "Stable Segmentations" is also superior to that of

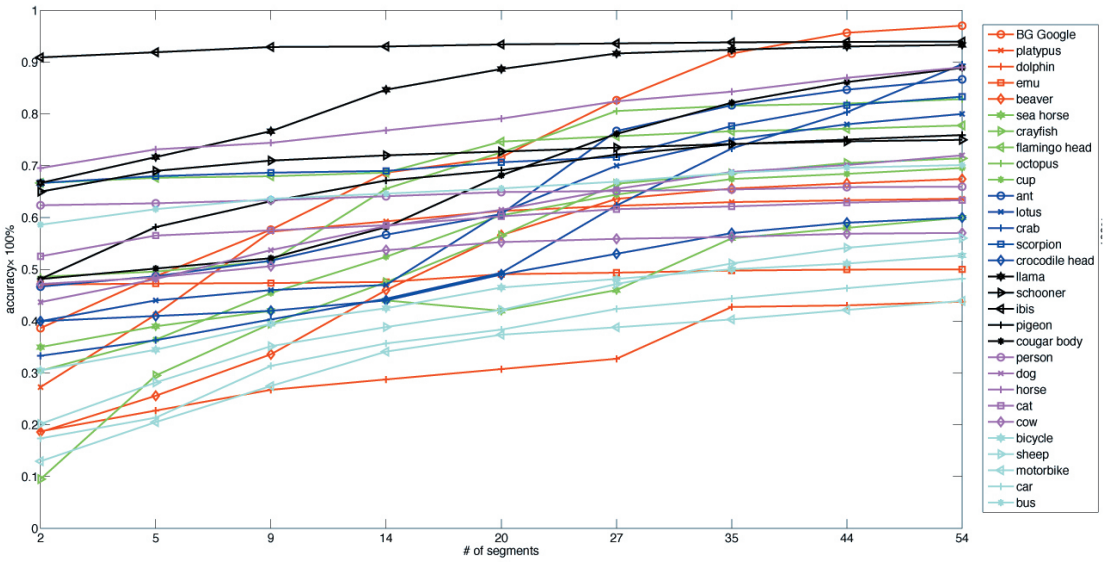


Figure 2.9: Object recognition accuracy vs. length of stable segmentations shortlist. Note the general trend of accuracy improvement as the number of segments increases. The accuracy improvement saturates at around 35 segments.

"Block Segmentations". The "Stable Segmentations", shown in Fig. 2.10 and 2.11, are capable of identifying objects' boundaries relatively accurately. Using "Block Segmentations", however, localization results are poor: the centroid of a segment often does not match the object center and segment boundaries truncate the objects.

Regardless of the particular segmentation algorithm, the size of the shortlist or the number of segments used to represent a test image can play an important role in determining object recognition accuracy. On one hand, as the number of possible segmentations increases, the chance of having a segment perfectly represent the object increases as well. On the other hand, an increase in the number of segments also increases the noise, namely, segments with incorrect category assignment. Figure 2.9 illustrates the effect of increasing the number of segments to represent the test images. The recognition accuracy of all categories significantly increases with the number of segments. However, around the 35 segment mark, the effect of the more accurate segment boundaries is cancelled out by the noise

Table 2.3: Average object localization accuracy for Caltech and PASCAL datasets. Accuracy > 0.5 constitutes a correctly localized object, according to the PASCAL 06 conventions.

Caltech	Bseg	Sseg
ant	0.24	0.47
BG Google	0.25	0.84
Beaver	0.23	0.65
Cougar body	0.27	0.69
Crab	0.27	0.51
Crayfish	0.24	0.53
Crocodile Head	0.37	0.72
Cup	0.31	0.77
Dolphin	0.31	0.78
Emu	0.19	0.64
Flamingo Head	0.17	0.62
Ibis	0.27	0.58
Llama	0.28	0.73
Lotus	0.40	0.65
Octopus	0.11	0.66
Pigeon	0.13	0.78
Platypus	0.19	0.71
Schooner	0.34	0.72
Scorpion	0.12	0.56
Sea Horse	0.16	0.62

Pascal	Bseg	Sseg
Bicycle	0.24	0.51
Bus	0.34	0.70
Car	0.34	0.66
Cat	0.26	0.62
Cow	0.30	0.64
Dog	0.23	0.60
Horse	0.26	0.52
Motorbike	0.14	0.43
Person	0.22	0.59
Sheep	0.19	0.67

from meaningless segments. Thus, for most categories, the recognition accuracy saturates past 35 segments per image (note that the 35 segments are distributed among 7 different segmentations).

2.5 Discussion

Although a link between image segmentation and object recognition has been discussed for many years, the effects of low-level global image segmentation on recognition and categorization have not been convincingly shown. In this work we demonstrated that image segmentation can in fact improve object recognition and categorization and it also adds object localization and multi-class categorization capabilities to an off-the-shelf categorization system.

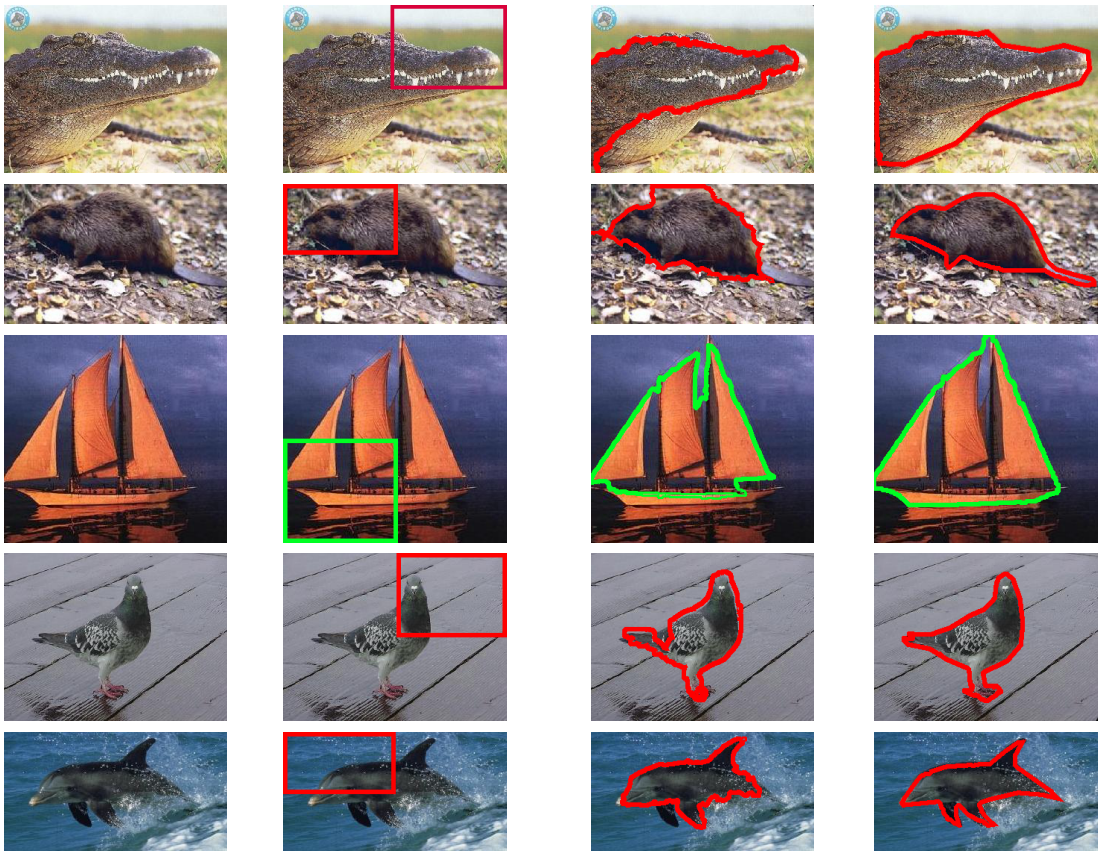


Figure 2.10: Object localization using "Stable Segmentations" as pre-processing for the BoF categorization model. Examples from the Caltech101 dataset. *Best viewed in color.*

Figure 2.11: Object localization using "Stable Segmentations" as pre-processing for the BoF categorization model. Examples from the PASCAL dataset. *Best viewed in color.*

Often segmentation has not been used in recognition because of the difficulty of obtaining segments corresponding to the objects of interest. In this chapter we solve this problem by relying on a shortlist of potentially meaningful segmentations (identified by a stability criterion), which significantly increase the chance of extracting suitable segments. Incorporating this segmentation method with a simple BoF model was shown to bring the recognition accuracy to a level comparable with the state-of-the-art [104], see Table 3.2.2.

We found that the quality of image segmentation does affect the average recognition accuracy for the BoF model. However, even the most trivial spatial grouping of interest points (i.e., Bseg) in the BoF model increases the categorization accuracy (but not as much as for Sseg). Localization is greatly affected by the segmentation quality as well. Segment boundaries from the "Stable Segmentations" approach often coincide closely with the object boundaries and object' and segments' centroids match, as shown in Figures 2.10 and 2.11. With "Block Segmentations", on the other hand, segment boundaries represent the object contours very poorly. The proposed approach of segmenting test images and recognizing individual segments allows for recognition of individual objects. Traditional approaches to recognition tend to suffer from multiple objects present in the scene. However, this model overcomes this issue by treating each segment independently of others. This method provides for an intuitive framework to explicitly capture the interactions between various segments and objects in the scene. In the next chapter, we build upon this framework and develop a contextual object recognition model.

Chapter 3.

Context

In the real world, objects tend to co-occur with other objects and particular environments, providing a rich collection of contextual associations to be exploited by the visual system. To take advantage of these associations, we extend the above described segmentation based BoF model by incorporating contextual interactions between objects in the scene. With object categorization in hand, a conditional random field (CRF) formulation is used as post-processing to maximize the objects' labels contextual agreement. It is important to note that contextual interactions between objects are captured by various statistical relations among objects instances. In addition to the appearance features described in the last chapter, this chapter introduces semantic, i.e., object co-occurrences, and spatial, i.e., geometric arrangement, contextual features into the object recognition model termed CoLA (Co-occurrence, Location, Appearance).

3.1 Segment Labeling Modified

As before, segments are classified based on a simple nearest neighbor rule. Define the un-normalized distance of the test segment S_q to class c as:

$$d(S_q, c) = \min_i d(S_q, I_{ic}) = \min_i \|\phi(S_q) - \phi(I_{ic})\|_1. \tag{3.1}$$

$d(S_q, c)$ is the minimum l_1 distance of the test segment S_q to all the training images I_{ic} of category c. We assign the segment S_q to its closest category $c_1(S_q)$:

$$c_1(S_q) = \underset{c}{\operatorname{argmin}}\, d(S_q, c). \tag{3.2}$$

Similarly, the S_q is assigned to the rest of the categories:
$c_i(S_q) = \text{sort}(d(S_q, c_i)), \forall\, 1 \leq i \leq n$, with sorting in ascending order of distance. In order to construct a probability distribution over category labels for image query segment, we introduce the following definition:

$$p(c_i|S_q) = \left[1 - \frac{d(S_q, c_i)}{\sum_{j=1}^{n} d(S_q, c_j)}\right]. \tag{3.3}$$

This definition of the labeling confidence is similar to the normalized exponential, or the *softmax* activation function. In the case of neural networks, where *softmax* was mainly applied, it was convenient to utilize the exponentials to represent the networks' scale parameters; however, the behavior of the exponentials is not suitable here. The labeling confidence is scaled by the segment importance: $p(c_i|S_q) = p(c_i|S_q)\omega(S_q)$. Thus, $p(c_i|S_q)$ is a probability distribution over category labels; it is proportional to the nearest neighbor distance between the query segment S_q and the category: $d(S_q, c)$. Given the labels of each segment, $c_i(S_q)$, all redundant segments (overlap $\geq 90\%$) are removed.

3.2 Semantic Context

To incorporate semantic context into the object categorization, we use a conditional random field (CRF) framework to promote agreement between the segment labels. CRFs have been widely used in object detection, labeling, and classification, see [31, 40, 60, 85]. The proposed CRF differs in two significant ways. First, we use a fully connected graph between segment labels instead of a sparse one. Second, instead of integrating the context model with the categorization

model, we train the CRF on simpler problems defined on a relatively small number of segments.

Context Model. Given an image I and its segments $S_1, \ldots, S_k$, we wish to find segment labels $c_1, \ldots, c_k$ such that they agree with the segment contents and are in contextual agreement with each other. We assume the labels come from a finite set $\mathcal{C}$.

We model this interaction as a probability distribution:

$$p(c_1 \ldots c_k | S_1 \ldots S_k) = \frac{B(c_1 \ldots c_k) \prod_{i=1}^{k} A(i)}{Z(\phi, S_1 \ldots S_k)}, \text{with} \tag{3.4}$$

$$A(i) = p(c_i|S_q) \text{ and } B(c_1 \ldots c_k) = \exp\Big(\sum_{i,j=1}^{k} \phi(c_i, c_j) \Big), \tag{3.5}$$

where $Z(\cdot)$ is the partition function. We explicitly separate the marginal terms $p(c|S)$, which are provided by the recognition system, from the interaction potentials $\phi(\cdot)$.

To incorporate semantic context information into object categorization, namely into the CRF framework, we construct context matrices. These are symmetric, nonnegative matrices that contain the co-occurrence frequency among object labels in the training set of the database (note that both MSRC and PASCAL databases have strongly labeled training data).

Co-occurence Counts. Our first source of data for learning $\phi(\cdot)$ is a collection of multiply labeled images $I_1, \ldots, I_n$. We indicate the presence or absence of label i with an indicator function l_i. The probability of some labeling is given by the model

$$p(l_1 \ldots l_{|\mathcal{C}|}) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i,j \in \mathcal{C}} l_i l_j \phi(i,j) \Big). \tag{3.6}$$

We wish to find a $\phi(\cdot)$ that maximizes the log likelihood of the observed label co-occurences. The likelihood of these images turns out to be a function only of the number of images, n, and a matrix of label co-occurence counts. An entry ij in this matrix counts the times an object with label i appears in a training image

with an object with label j. The diagonal entries correspond to the frequency of the object in the training set. The structure and content of these matrices for MSRC and PASCAL training datasets is illustrated Figure 3.1(3rd column).

It is intractable to maximize the co-occurence likelihood directly, since we must evaluate the partition function to do this. Hence, the partition function is approximated using Monte Carlo integration, as in [77]. Importance sampling is used where the proposal distribution assumes that the label probabilities are independent with probability equal to their observed frequency. Every time the partition function is estimated, $40,000$ points are sampled from the proposal distribution.

We use simple gradient descent to find a $\phi(\cdot)$ that approximately optimizes the data likelihood. Due to noise in estimating Z, it is hard to check for convergence; instead training is terminated when 10 iterations of gradient descent do not yield average improved likelihood over the previous 10.

3.2.1 Sources of Semantic Context on Object Recognition

In practice, most image databases do not have a training set with an equal semantic context prior and/or strongly labeled data. Thus, we would like to be able to construct $\phi(\cdot)$ from a common knowledge base, obtained from the Internet. In particular, we wish to generate contextual constraints among object categories using Google Sets[1] (GS).

Google Sets generates a list of possibly related items, or objects, from a few examples. It has been used in linguistics, cell biology and database analysis to enforce contextual constraints [24, 73, 82]. In order to obtain this information for object categorization we queried Google Sets using the labeled training data available in the MSRC and PASCAL databases. We generated a query using every category label (one example) and then matched the results against all the categories present in these datasets. This task was performed for each database

[1] http://labs.google.com/sets

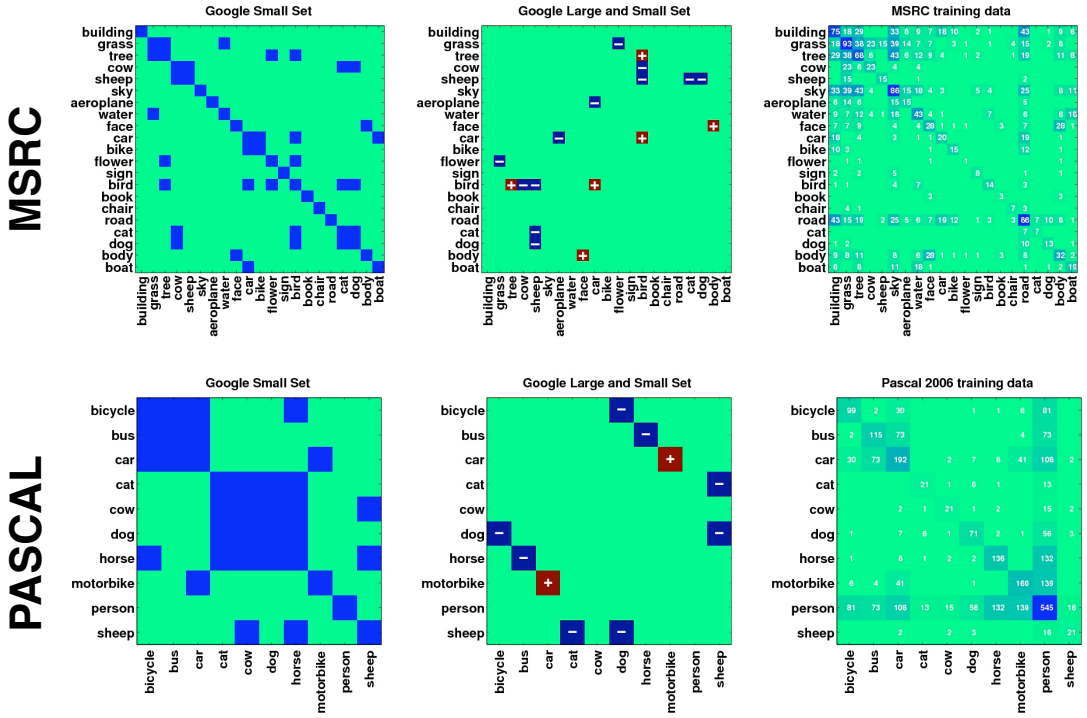


Figure 3.1: Context matrices for MSRC and PASCAL datasets. **Google Small Set:** Binary context matrix from GS_s. Blue pixels indicate a contextual relationship between categories. **Google Large and Small Set:** Differences between small and large Google Sets context matrices. '-' signs correspond to relations present GS_s but not in GS_l; '+' correspond to relations present GS_l but not in GS_s. **Training Data:** Ground Truth, training set label co-occurence, context matrix.

using the small set, GS_s containing 15 results, and the large set, GS_l, comprised of upto 60 entries. Figure 3.1 (left column) show binary contexts from GS_s, for MSRC and PASCAL respectively. Intuitively, we expected $GS_S \subset GS_l$, however, $GS_s \setminus GS_l \neq \emptyset$ as shown in Figure 3.1 (middle column). The larger set implies broader relations, thus changing the context of the set to be too general. In this work we retrieve objects labels' semantic context from GS_s.

In this case, $\phi(i,j) = \gamma$ if GS_s marks them as related, or 0 otherwise. We set $\gamma = 1$ for our experiments, though γ could be chosen using cross-validation on training data if available.

Besides Google Sets, we considered other sources of contextual information such as WordNet [18] and Word Association[2]. In the task of object categorization we found that these databases did not offer sufficient semantic context information for the visual object categories, either due to the limited recall (in Word Association) or irrelevant interconnections (in Wordnet).

3.2.2 Effects of Semantic Context

As mentioned earlier, we are interested in a relative performance change in object categorization accuracy, i.e., with and without post-processing with semantic context. In Table 3.2.2 we summarize the performance of average categorization accuracy for both the MSRC and PASCAL datasets. These results are competitive with the current state-of-the-art approaches [85, 104]. The confusion matrices, which describe the results in more details, are shown in Figure 3.2. For both datasets the categorization results improved considerably with inclusion of context. For the MSRC dataset, the average categorization accuracy increased by more than 10% using the semantic context provided by Google Sets, and by over 20% using the ground truth training context. In the case of PASCAL, the average categorization accuracy improved by about 2% using Google Sets, and by over 10% using the ground truth. In Figure 3.9 are examples where context improved object categorization. In examples 1 and 3, semantic context constraints help correct an entirely wrong appearance based labeling: bicycle – boat, and boat – cow. In examples, 2,4,5 and 6, mislabeled objects are visually similar to the ones they are confused with: boat – building, horse – dog, and dog – cow. Thus, it seems that contextual information may not only help disambiguate between visually similar objects, but also correct for erroneous appearance representation.

Unfortunately, context constraints can also lower or leave the categorization accuracy unchanged. As shown in Figure 3.10, the initially correct labels,

[2]http://www.wordassociation.org

"building" in the first image, and "grass" in the second, were relabeled incorrectly in favor of semantic context relations learned from the co-occurences in the training data. Most of such mistakes are due to the initial probability distribution over labels, $p(c|S_q)$; the feature description is not very rich as the SIFT descriptor used in this work is color-blind and segment shapes are only captured implicitly. In combining our approach with a method of strong feature description, e.g., [85], many of currently encountered errors will likely be eliminated. As was noted by [7], there are other types of context. Relative scale, spatial arrangements of objects and other types of object statistics are also of great importance for object recognition. Although we do not consider them all, in the next section we include spatial context and show how to modify the existing recognition model to incorporate a variety of contextual constraints.

Table 3.1: Average categorization accuracy with and without semantic contextual constraints. Context dependencies are learned either from Google Sets or from training data.

	No Context	Google Sets	Using Training
MSRC	45.0%	58.1%	68.4%
PASCAL	61.8%	63.4%	74.2%

3.3 Spatial Context

Since semantic context is not the only source of contextual information for object recognition, it is important to develop a model that is capable of incorporating multiple facets of contextual data. Here, we append semantic context with spatial arrangements among objects and modify the existing model of object recognition, Equation 3.5, to include a variety of contextual constraints.

As before, given an image I, its corresponding segments $S_1, \ldots, S_k$, and probabilistic per segment labels $p(c_i|S_q)$, we wish to find segment labels $c_1, \ldots, c_k \in \mathcal{C}$ such that all agree with the segments' content and are in contextual agreement

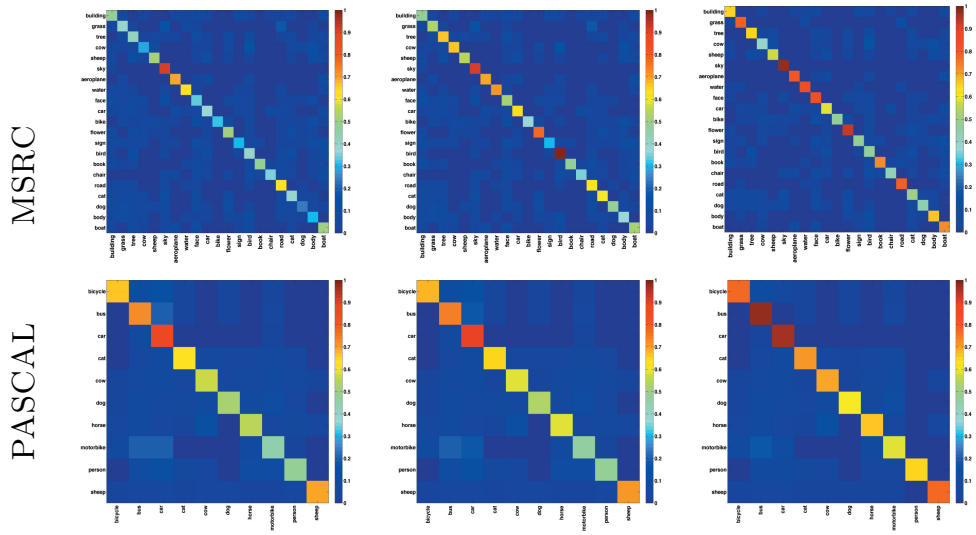


Figure 3.2: Confusion matrices of average categorization accuracy for MSRC and PASCAL datasets. First row: MSRC dataset; second row: PASCAL dataset. (a) Categorization with no contextual constraints. (b) Categorization with Google Sets context constraints. (c) Categorization with Ground Truth context constraints learning from training data.

with one other. We model this interaction as a probability distribution:

$$p(c_1 \dots c_k | S_1 \dots S_k) = \frac{B(c_1 \dots c_k) \prod_{i=1}^{k} p(c_i | S_q)}{Z(\phi_0, \dots \phi_r, S_1 \dots S_k)}, \tag{3.7}$$

$$\text{with } B(c_1 \dots c_k) = \exp\Big(\sum_{i,j=1}^{k} \sum_{r=0}^{q} \alpha_r \phi_r(c_i, c_j) \Big), \tag{3.8}$$

where $Z(\cdot)$ is the partition function and q is the number of pairwise spatial relations. To incorporate both semantic and spatial context information into the CRF framework, we construct context matrices as described next.

Location. Spatial context is captured by co-ocurrence matrices for each of the four pairwise relationships (*above*, *below*, *inside* and *around*), as shown in Figure 3.4. These high level labels are meant to provide an intuition of geometric interaction between objects. The actual spatial descriptor simply vector quantizes the geometric features, as described in the next section.

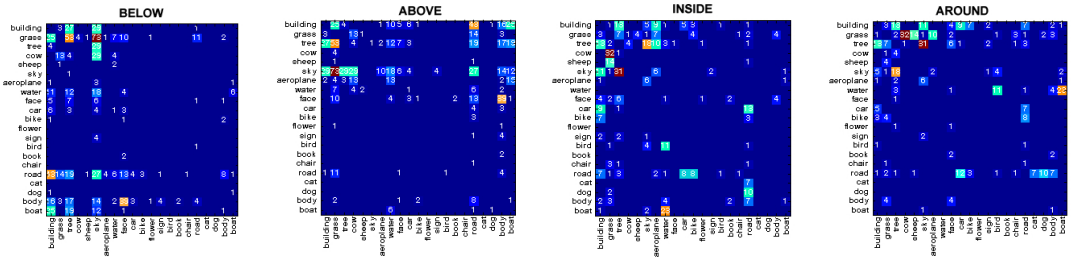


Figure 3.3: Source of contextual information. Co-occurrence matrices for spatial relationships above, below, inside and around for MSRC database. Each entry ij in a matrix counts the number times an object with label i appears in a training image with an object with label j according to a given pairwise relationship.

The matrices contain the frequency among objects labels in the four different configurations, as they appear in the training data. An entry (i, j) in matrix $\phi_r(c_i, c_j)$, with $r = 1, \ldots, 4$, counts the number of times an object with label i appears with an object label j for a given relationship r. Figure 3.3 illustrates the counts over the four different relationships for MSRC and PASCAL. It is worth noting that MSRC matrices exhibit more uniform interactions between objects, while matrices of PASCAL single out categories of very high activity (e.g., *person*).

Co-occurrence Counts. While the co-occurrences of category labels are captured by the spatial context matrices above, the appearance frequency – a parameter required for the CRF – is not captured explicitly, since these matrices, Figure 3.3, are hollow. Using the existing context matrices, object appearance frequency can be computed as row sums of all four matrices. Finally, the sum of all the matrices, including the row sums, will result in a marginal (i.e., without regard for location) co-occurrence matrix, equivalent to those presented in Figure 3.1. An entry (i, j) in the semantic context matrix counts the times an object with label i appears in a training image with an object with label j. The diagonal entries correspond to the frequency of the object in the training set:

$$\phi_0(c_i, c_j) = \phi'(c_i, c_j) + \sum_{k=1}^{|\mathcal{C}|} \phi'(c_i, c_k) \tag{3.9}$$

where $\phi'(\cdot) = \sum_{r=1}^{q} \phi_r(c_i, c_j)$. Therefore the probability of some labeling is given

by the model

$$p(l_1 \dots l_{|\mathcal{C}|}) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i,j \in \mathcal{C}} \sum_{r=0}^{q} l_i l_j \cdot \alpha_r \cdot \phi_r(c_i, c_j) \Big), \tag{3.10}$$

with l_i indicating the presence or absence of label i.

3.3.1 Sources of Spatial Context

[8] proposed that physical and semantic changes in a coherent scene interfere with and cause delays in object recognition. Conversely, object recognition can be facilitated by the use of relationships that support the definition of a coherent scene.

In the area of object recognition and scene understanding, several works have incorporated the use of spatial relationships as a source of context. The work of Singhal et al. [87] combines probabilistic spatial context models and material detectors for scene understanding. These models are based on pre-defined pixel level relationships between image regions, where spatial context information is represented as a binary feature of each specified relationship. [41] model interactions among pixels, regions and objects using a hierarchical CRF. In their model, the computed regions and objects are a result of the CRF itself. Although it is possible to capture a variety of different low level pixel groupings in the first level of their hierarchy, the authors only consider a single equilibrium configuration and propagate it (along with its uncertainty) to the level of regions and objects.

In contrast, our method employs a decoupled segmentation stage that extracts a shortlist of stable (and possibly overlapping) segments as input to a subsequent context based reasoning stage. As a result, the latter stage – also CRF-based – has at its disposal a variety of shortlists of possible objects and labels over which to perform inference based on co-occurrence and spatial relationships. These relationships, which in our case are unknown *a priori*, characterize the nature of object interaction in real world images and reveal important information to disambiguate

object identity. It is important to note, that object statistics such as location, scale, and others, should be captured in the relation to the other objects in the scene, rather than on the absolute scale of the image. Our sources of information

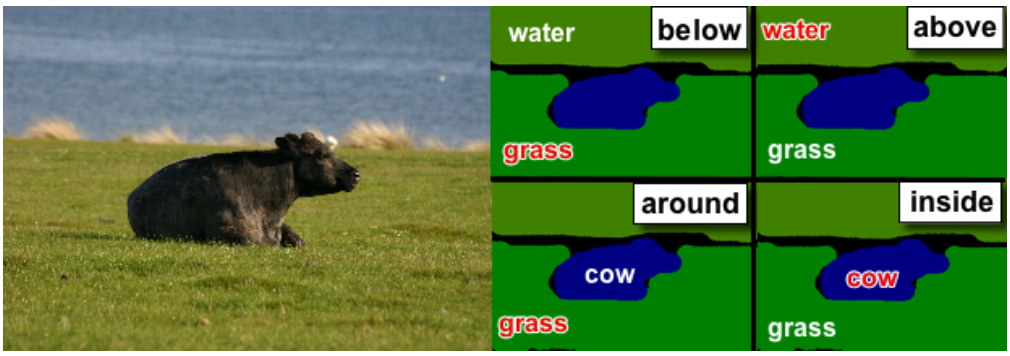


Figure 3.4: Illustration of four basic spatial relationships that exist among objects within an MSRC image. Labels in red indicate the object that possesses the relationship with respect to the object with the white label, e.g, the grass, in red, is below water, in white.

for learning spatial configurations on pairs of objects are the MSRC and PASCAL training databases. In particular, these sources provide us a collection of multiply labeled images $I_1, \ldots, I_n$, each containing at least two objects belonging to different categories, $c_i, c_j \in \mathcal{C}$ s.t. $i \neq j$; an object i is labeled by a bounding box or pixel mask β_i. We define the following simple pairwise feature to capture a specific object configuration as a three dimensional spatial context descriptor:

$$F_{ij} = (\mu_{ij}, O_{ij}, O_{ji})^\top \ \forall i, j \in \mathcal{C},\ i \neq j, \tag{3.11}$$

$$O_{ij} = \frac{\beta_i \cap \beta_j}{\beta_i \cup \beta_j} \text{ and } \mu_{ij} = \mu_{yi} - \mu_{yj} \tag{3.12}$$

where μ_{ij} is the difference between the y component of the centroids (in normalized coordinates) of the objects labeled c_i and c_j, and O_{ij} is the overlap percentage of the object with label c_j with respect to the object with label c_i. We omit the x component of the centroid since relative horizontal position does not carry any discriminative information for the objects in PASCAL or MSRC. We specifically

designed the spatial descriptor to be simple for computational efficiency and to be consistent with motivations in vision science, as discussed in [69].

In order to capture the prevalent spatial arrangements among objects in the databases, we vector quantize the feature space into 4 groups. Choosing a small number of groups translates into simpler relations that can explain interactions that are well represented across many object pairs and scenes. We used the ground truth segmented regions and bounding boxes labels from MSRC and PASCAL 2007, respectively, to compute the spatial context descriptors. A closer look at the resultant clusters, shown in Figure 3.5, suggests the pairwise relationships *above*, *below*, *inside* and *around*, illustrated for an example image in Figure 3.4 containing *grass*, *water* and *cow*. Learning the relationships between pairs of objects, rather than defining them *a priori*, yields a more generic and robust description of spatial interactions among objects.

Despite the differences between MSRC and PASCAL datasets, the distributions we observe in Figure 3.5 have comparable overall shapes, and the clusters representing the spatial relations are found in similar locations in the feature space. In the case of MSRC, the *above* and *below* relationships are predominant, as many objects remain in vertically consistent locations relative to other objects (e.g., sky, water, grass). In contrast, PASCAL's biggest clusters correspond to the spatial relationships *inside* and *around*, since most of these objects are found interposed with respect to one another. Also, as PASCAL object labels are specified by bounding boxes, rather than pixel-resolution ground truth masks, the average overlap values are thus greater.

3.3.2 Empirical Effects of Inclusion of Spatial Context

To evaluate categorization accuracy of the proposed model and the relative importance of spatial context in this task, we consider MSRC and PASCAL 2007 datasets. Table 3.2 summarizes the performance of average categorization per

category.

Table 3.2: Comparison of recognition accuracy between the models for MSRC and PASCAL categories. Results in **bold** explain an increase in performance by our model. A decrease in performance is shown in *italics*.

categories	semantic context	CoLA
building	0.85	**0.91**
grass	0.94	**0.95**
tree	0.78	**0.80**
cow	0.36	**0.41**
sheep	0.55	0.55
sky	0.89	**0.97**
aeroplane	0.73	0.73
water	0.95	0.95
face	0.80	**0.81**
car	0.57	0.57
bike	0.59	**0.60**
flower	0.65	0.65
sign	0.54	0.54
bird	0.54	*0.52*
book	0.56	0.56
chair	0.42	0.42
road	0.94	**0.96**
cat	0.42	0.42
dog	0.46	0.46
body	0.75	**0.77**
boat	0.76	**0.81**

categories	semantic context	CoLA
aeroplane	0.63	0.63
bicycle	0.22	0.22
bird	0.18	*0.14*
boat	0.28	**0.42**
bottle	0.43	0.43
bus	0.46	**0.50**
car	0.62	0.62
cat	0.32	0.32
chair	0.37	0.37
cow	0.19	0.19
diningtable	0.30	0.30
dog	0.32	*0.29*
horse	0.12	**0.15**
motorbike	0.31	0.31
person	0.43	0.43
pottedplant	0.33	0.33
sheep	0.41	0.41
sofa	0.37	0.37
train	0.29	0.29
tvmonitor	0.62	0.62

These results are at par with current state-of-the-art approaches, [17, 86], and in some categories exhibit much improved recognition accuracy. In particular, the average categorization per database, is 68.38% for MSRC and 36.7% for PASCAL. What is of more interest to us, however, is the per category accuracy as a function of the type of context used. Specifically, we notice that around half of the 21 categories in MSRC benefit from using spatial context: an increase from 1% up to 8% in recognition accuracy. For the rest of the categories, in turn, spatial context did not harm the performance, except for a small decrease in accuracy on category *bird*.

In the PASCAL database, the availability of spatial context data is less uniform across categories. An increase is seen in only three categories, though in

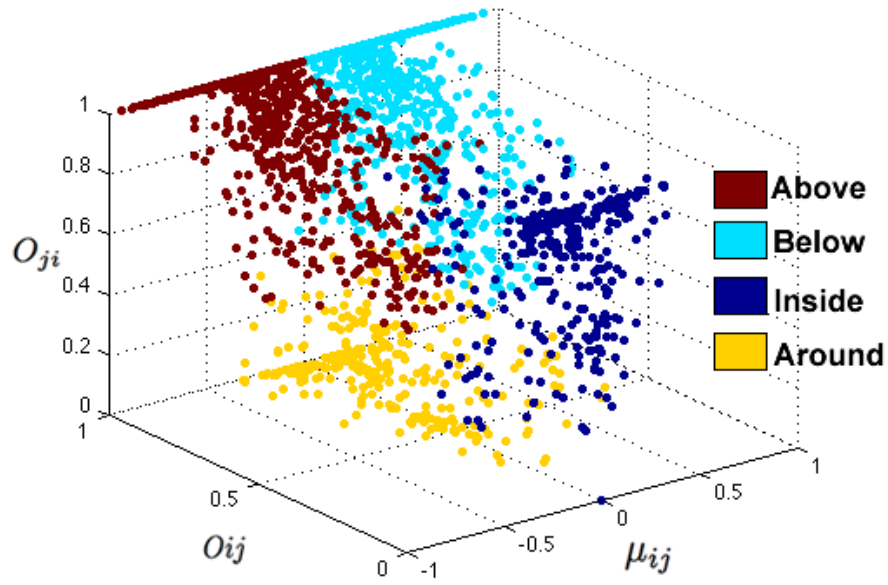
one case (for category *boat*) this increase was rather significant (14%). As with MSRC, the other categories are largely unaffected by spatial context, and only one category (*bird*) suffers from reduced accuracy.

Figure 3.6 summarizes the relative improvement of categorization accuracy with the inclusion of spatial context into the recognition model. Very few categories' recognition accuracy is worsened by spatial context; most are either unchanged or improved. Some examples of affected categories are shown in Figures 3.7 and 3.8.

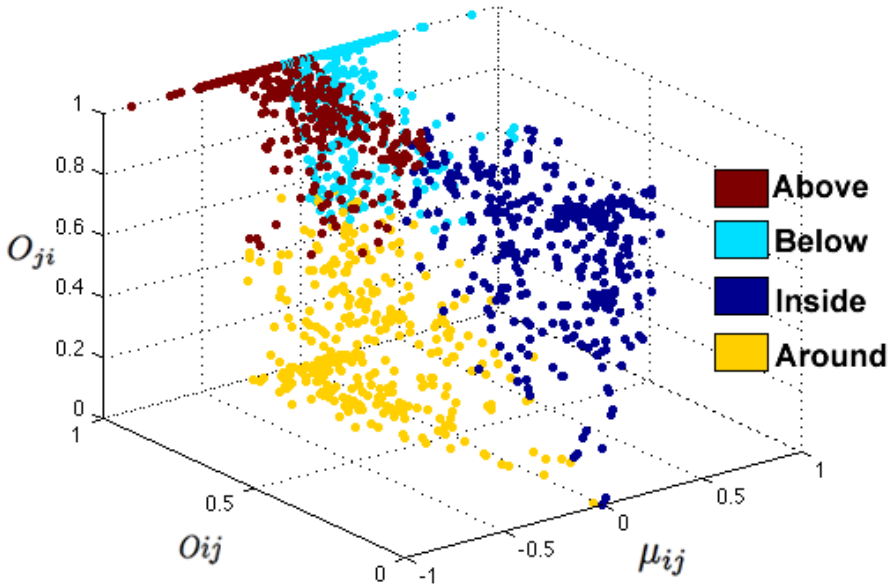
Run Time and Implementation Details. All test images were represented by multiple stable segmentations that were implemented with normalized cuts [84, 12], using brightness and texture cues, as was originally presented by [74]. We considered 9 segmentations per test image, where the number of segments per segmentation ranges from $k = 2, \ldots, 10$. The computation time for each segmentation is between 10-20 seconds per image. As the individual segmentations are independent of one another, we computed them all in parallel. As a result, a computation of all stable segmentations per image requires about 10 minutes.

15 and 30 training images were used for the MSRC and PASCAL databases respectively. 5000 random patches at multiple scales (from 12 pixels up to the image size) are extracted from each image. The feature appearance is represented by SIFT descriptors, [51], and the visual words are obtained by quantizing the feature space using hierarchical K-means with $K = 10$ at three levels , [64]. The image signature is a histogram of such hierarchical visual words, L_1 normalized and TFxIDF re-weighed, [64]. The computation of SIFT and the relevant signature, implemented in C, takes on average 1.5 seconds per segment. Training and constructing the vocabulary tree requires less than 40 minutes for 20 categories with 30 training images in each category, in the case of PASCAL. Classification of test images is done in just a few seconds. Training the CRF takes 3 minutes for 315 training images for MSRC and 5 minutes for 600 images in PASCAL training

dataset. Enforcing semantic and spatial constraints on a given segmentation takes between 4-7 seconds, depending on the number of segments. All the above operations were performed on a Pentium 3.2 GHz.



(a)



(b)

Figure 3.5: Four different groups represent four different spatial relationships: *above*, *below*, *inside* and *around*. For MSRC we observe many more pairwise relationships that belong to vertical arrangements. For PASCAL 2007 we observe comparatively more pairwise relationships that belong to overlapping arrangements.

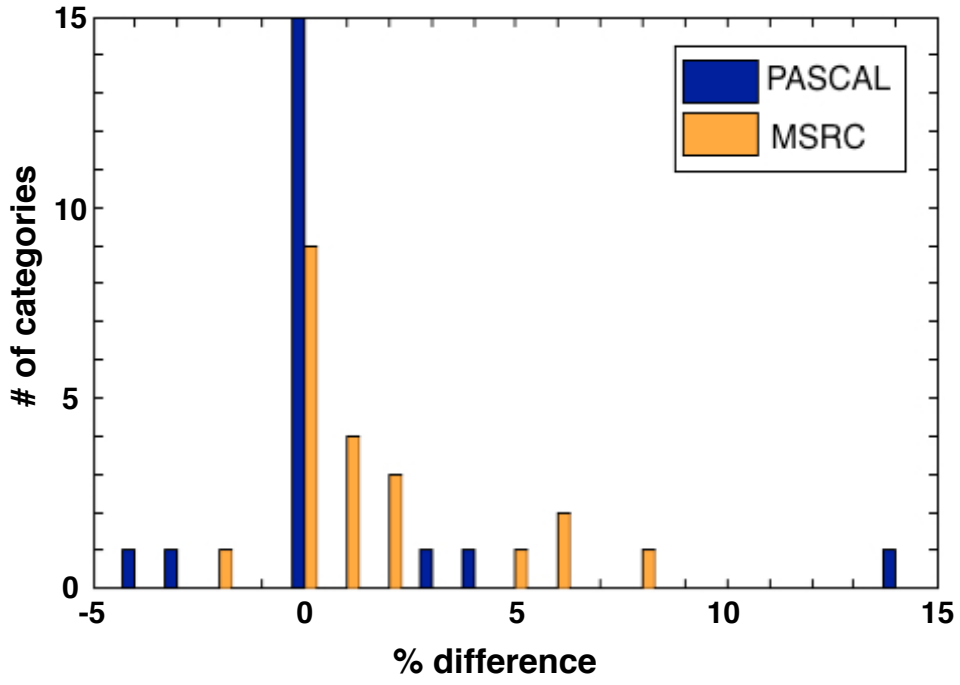
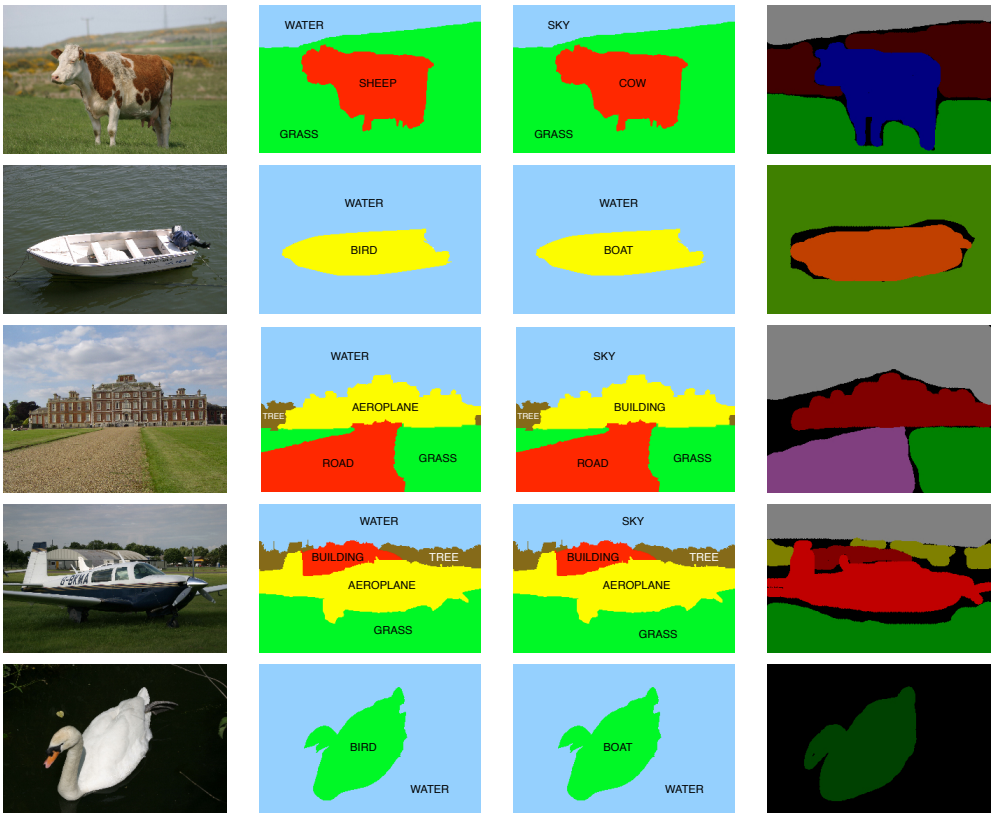


Figure 3.6: Difference in performance between semantic and semantic+spatial framework for MSRC and PASCAL databases.

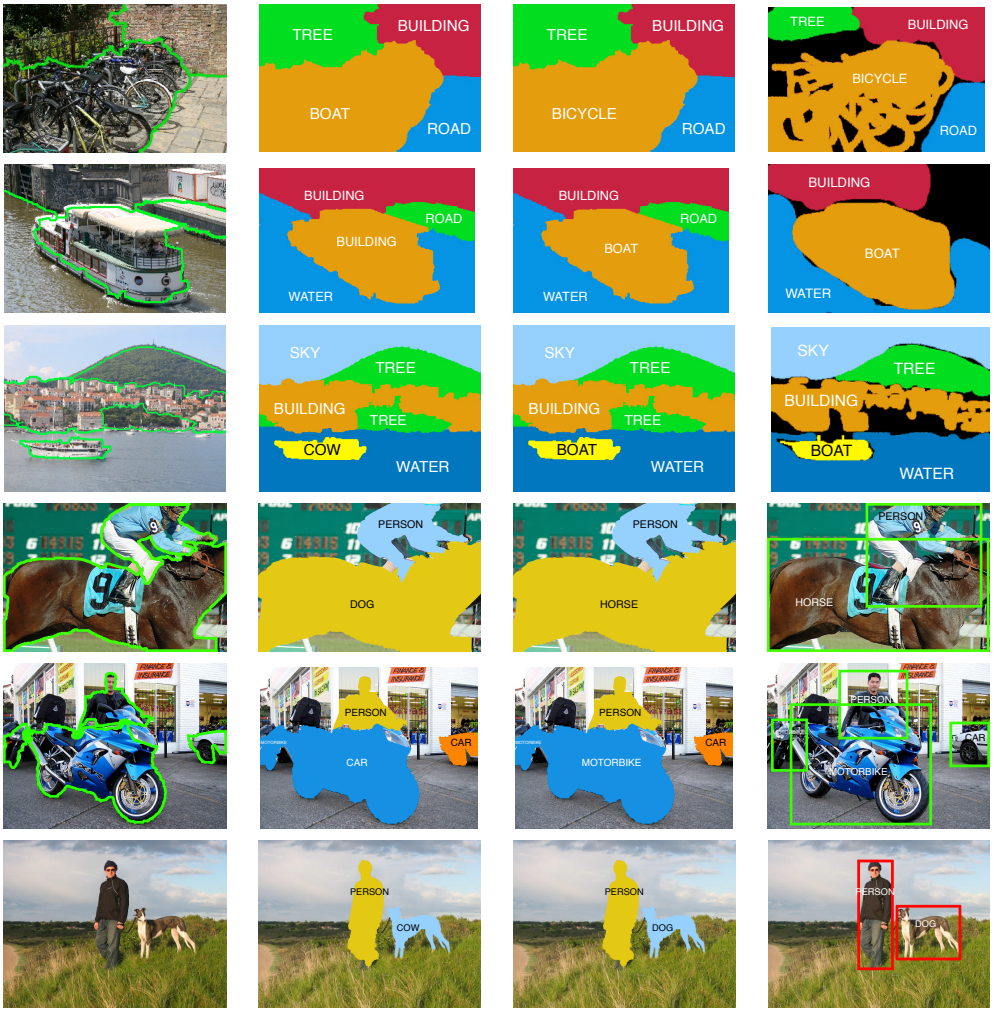


(a) (b) (c) (d)

Figure 3.7: Examples of images from the MSRC database. Spatial constraints have improved (first four rows) and worsened (last row) the categorization accuracy. Full segmentations of highest average categorization accuracy are shown. (a) Original image. (b) Categorization with co-occurrence contextual constraints. (c) Categorization with spatial and co-occurence contextual constraints.(d) Ground Truth.

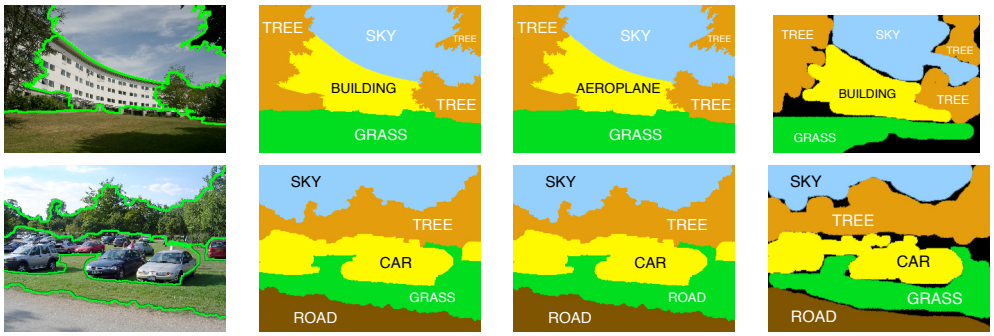
(a) (b) (c) (d)

Figure 3.8: Examples of images from the PASCAL 07 database. Spatial constraints have improved (first four rows) and worsened (last row) the categorization accuracy. Individual segments of highest categorization accuracy are shown. (a) Original image. (b) Categorization with co-occurrence contextual constraints. (c) Categorization with spatial and co-occurence contextual constraints.(d) Ground Truth.



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

Figure 3.9: Examples of MSRC (first 3) and PASCAL (last 3) test images, where contextual constraints have improved the categorization accuracy. Results are shown in two different ways, one for each dataset. In MSRC, the consensus segmentation is shown to match the style of the ground truth; in PASCAL individual segments of highest categorization accuracy are shown since only few segments have high enough confidence of being a particular category, and thus are shown. Many object categories that are found in the images (i.e., sky, grass, building) are not part of the training set in PASCAL, thus labeling of those segments becomes random. (a) Original Segmented Image. (b) Categorization without contextual constraints. (c) Categorization with co-occurence contextual constraints derived from the training data. (d) Ground Truth.



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

Figure 3.10: Examples of MSRC test images, where contextual constraints have reduced the categorization accuracy. (a) Original Segmented Image. (b) Categorization without contextual constraints. (c) Categorization with co-occurence contextual constrains derived from training data. (d) Ground Truth Categorization.

Chapter 4.

Contextual Modeling in Object Recognition

In the computer vision community, contextual models for object recognition were introduced in late 1980's and early 1990's [21, 58, 88], and were popularized by Oliva and Torralba in 2001 [66]. Although with different formulations, most of the approaches can be classified into two general categories: (i) models with contextual inference based on the statistical summary of the scene (we will refer to these as scene based context models, or SBC), and (ii) models representing the context in terms of relationships among objects in the image (object based context, or OBC).

The approach of [66], later termed Gist [91], was fundamental among the SBC models. Since then, variants of the SBC model were presented in [31, 41, 98, 102]. These recent works have shown that a statistical summary of the scene provides a complementary and effective source of information for contextual inference, which enables humans to quickly guide their attention to regions of interest in natural scenes.

SBC models of context, Gist-based approaches in particular, aim to capture the surrounding information around the object of interest. By incorporating the

statistics of the clutter or background, context becomes a global feature of the object category. For example, refrigerators usually appear in a kitchen, thus the usual background of refrigerators is similar. Having learned such a global feature of an object category, one can infer a potential object label: if the background resembles a kitchen, then the patch of interest may be a refrigerator. However, many objects can have similar backgrounds, e.g., refrigerators, coffee makers, and stoves all belong in the kitchen. Alternatively, instances of a particular object (a face or a car), may have very different backgrounds depending on the environment they are in. Faces, for example, may appear outdoors or inside, at night or during the day. As illustrated in Figure 4.1(a,c), the background of an object may not always be indicative of the object itself.

Proceeding with the SBC model, after measuring the global features of the image, one first infers the scene context of the image, e.g., kitchen, and then with scene context in hand, the label of the object is inferred, e.g., refrigerator. Notice that if the scene context is inferred incorrectly, it becomes impossible to identify the object label accurately.

An alternative approach to Gist and other SBC models is to use a method based on the OBC model, variant of which was presented in Section 3.2. Rather than measuring global image statistics, inter-object constraints are imposed on potential object candidates in the image. With learned category interaction probabilities, either from training data or generic sources on the web, object labels are given to image regions, such that mutual co-occurrence and spatial constraints among all the object labels in the image are maximized. In OBC approaches, only the object category labels must be inferred given the context between categories and individual object appearance, without regard for scene context. To illustrate this further, return to the example of an idealized OBC model in Figure 1.1.

In the scene of a tennis match, four objects are detected and categorized: “Tennis court”, “Person”, “Tennis Racket”, and “Lemon”. Using a categorization system without a contextual module, these labels would be final; however, in con-

(a) (b) (c)

Figure 4.1: The structure of objects and their backgrounds (taken from [81]). In this illustration, each image has been created by averaging hundreds of images containing a particular object in the center (a face, keyboard and fire hydrant) at a fixed scale and pose. Before averaging, each image is translated and scaled so that the target object is in the center. The averages can reveal the regularities existing in the color/brightness patterns across all the images. However, this behavior is only visible for the keyboard in (b). In (a), the background of a face is approximately uniform, since faces appear in a variety of settings. Alternatively in (c), the background of a fire hydrant, may be identical to that of a bus stop of a street sign.

text, one of these labels is not satisfactory. Namely, the object labeled "Lemon", with an appearance very similar to a "Tennis Ball" is mislabeled due to the ambiguity in visual appearance. By modeling context with OBC constraints provided by an oracle, the label of the yellow blob changes to "Tennis Ball," as this label better satisfies the contextual conditions. While the above mentioned formulations of context appear rather different, it is clear that inclusion of context, in some form, in object recognition is a must. Thus, we are faced with a dilemma: which contextual model is more suitable in the framework of automated object recognition or categorization? Furthermore, which model is simpler, and finally, do the differences in the formulations matter? In the following sections, we formulate both SBC and OBC models in a manner most suitable for a direct comparison and an evaluation.

4.1 Scene Based Context (SBC) Model

To provide the necessary analysis of SBC models we pick a representative formulation of Gist. To stay consistent with the original work, we will use the same notation as in [91].

Consider an image with image statistics represented by some measurement $\mathbf{v}$. In particular, let $\mathbf{v} = \{\mathbf{v}_L, \mathbf{v}_C\}$, where $\mathbf{v}_L$ refers to statistics in the local spatial neighborhood, at scale σ, around some interest point at location x; $\mathbf{v}_L = \{\sigma, x\}$. $\mathbf{v}_C$ captures the image statistics from the rest of the image (contextual information); $\mathbf{v}_C$ is a low dimensional holistic representation that encodes the structural scene information. In other words, there is a *correlation* between low level representation of the scene and the objects that can be found inside. A typical appearance based object likelihood function $p(O|\mathbf{v}) = \frac{p(O,\mathbf{v})}{p(\mathbf{v})}$, with O being the object of interest, can now be re-written as $p(O|\mathbf{v}) = p(O|\mathbf{v}_L, \mathbf{v}_C)$. It is important to note that majority of the existing approaches to recognition simply omit $\mathbf{v}_C$, and only compute $p(O|\mathbf{v}_L)$. To formally include the contextual information into the objective function, we use Bayes' rule to re-write (1):

$$p(O|\mathbf{v}) = \frac{p(O,\mathbf{v})}{p(\mathbf{v})} = \frac{p(\mathbf{v}_L|O,\mathbf{v}_C)p(O|\mathbf{v}_C)}{p(\mathbf{v}_L|\mathbf{v}_C)} \quad (4.1)$$

$$= \frac{p(\mathbf{v}_L|O,\mathbf{v}_C)p(O|\mathbf{v}_C)}{p(\sigma,x|\mathbf{v}_C)} = \frac{p(\mathbf{v}_L|O,\mathbf{v}_C)p(O|\mathbf{v}_C)}{p(\sigma|x,\mathbf{v}_C)p(x|\mathbf{v}_C)}, \quad (4.2)$$

where $p(\mathbf{v}_L|O,\mathbf{v}_C)$ refers to the spatial relationship between objects: knowing the object label O, and the context of the scene $\mathbf{v}_C$, what is the most probable location of the object in such an image; $p(\sigma|x,\mathbf{v}_C)p(x|\mathbf{v}_C)$ is the normalization term referring to the distribution of scales and locations for various contexts; and finally $p(O|\mathbf{v}_C)$ is the contextual object recognition term.

Let us concentrate on $p(O|\mathbf{v}_C)$. The object label O incorporates the scale at which the object is found, the label, and the location in the image: $O = \{\sigma, o, x\}$. The function of interest here, $p(O|\mathbf{v}_C)$, can thus be factored as:

$$p(O|\mathbf{v}_C) = p(\sigma|x,o,\mathbf{v}_C)p(x,|o,\mathbf{v}_C)p(o|\mathbf{v}_C), \quad (4.3)$$

where $p(\sigma|x, o, \mathbf{v}_C)$ is the scale selection component, $p(x|o, \mathbf{v}_C)$ is the focus of attention (i.e., the most likely location for the object of interest) and $p(o|\mathbf{v}_C)$ is the contextual priming. This function is further evaluated in [91]. Here, however, by the chain rule of conditional probability, $p(O|\mathbf{v}_C)$ can be decomposed in a number of different ways. For example:

$$p(O|\mathbf{v}_C) = p(o|\sigma, x, \mathbf{v}_C)p(\sigma|x, \mathbf{v}_C)p(x|\mathbf{v}_C), \tag{4.4}$$

where $p(o|\sigma, x, \mathbf{v}_C)$ is the contextual priming given context, object location and scale, $p(\sigma|x, \mathbf{v}_C)$ is the scale parameter, and $p(x|\mathbf{v}_C)$ determines the most probable location of the object in the image.

In turn, let's examine $p(o|\sigma, x, \mathbf{v}_C)$ in detail. The label of the object is dependent on its physical properties (σ and x), and its surroundings ($\mathbf{v}_C$). Furthermore, it is generally true that physical properties of objects are independent of context: $(x, \sigma) \perp \mathbf{v}_C$. For example, a human face may be of different sizes and may appear in different locations in the image, independent of the context that it is in. Therefore, it is reasonable to assume that if scale and position are independent of context given the object label, then $p(\sigma, x, \mathbf{v}_C|o) = p(\sigma, x|o)p(\mathbf{v}_C|o)$. In turn, $p(o|\sigma, x, \mathbf{v}_C) = \frac{p(o|\sigma,x)p(o|\mathbf{v}_C)}{p(o)}$, since $p(o)$ is constant (i.e., same number of training images per category), this term is omitted for clarity. Thus, we can re-write (2) as follows:

$$\begin{aligned} p(O|\mathbf{v}) &= \frac{p(\mathbf{v}_L|O, \mathbf{v}_C)p(o|\sigma, x, \mathbf{v}_C)p(\sigma|x, \mathbf{v}_C)p(x|\mathbf{v}_C)}{p(\sigma|x, \mathbf{v}_C)p(x|\mathbf{v}_C)} && (4.5) \\ &= p(\mathbf{v}_L|O, \mathbf{v}_C)p(o|\sigma, x)p(o|\mathbf{v}_C). && (4.6) \end{aligned}$$

For the multi object case

$$p(o_n|\mathbf{v}_C) = \sum_{i=1}^{k} p(o_n|C_i, \mathbf{v}_C)p(C_i|\mathbf{v}_C) \approx \sum_{i=1}^{k} p(o_n|C_i)p(C_i|\mathbf{v}_C), \tag{4.7}$$

where k is the number of possible scenes, C_i are various scene context categories, and o_n is the label for the nth object. Finally:

$$p(O_n|\mathbf{v}) = p(\mathbf{v}_L|O_n, \mathbf{v}_C)p(o_n|\sigma, x) \sum_{i=1}^{k} p(o_n|C_i)p(C_i|\mathbf{v}_C). \tag{4.8}$$

In this approach, the statistics of the local neighborhood $\mathbf{v}_L$ and the contextual information $\mathbf{v}_C$ are both represented using global image features. In particular, in the scene representation proposed in [66], the image is first decomposed by a bank of multiscale oriented filters (tuned to eight orientations and four scales). Then, the output magnitude of each filter is averaged over 16 non-overlapping windows arranged on a 4 grid. The resulting image representation is a $4 \times 8 \times 16 = 512$ dimensional vector. The final feature vector, used to represent the entire image, is obtained by projecting the binned filter outputs onto the first 80 principal components computed on a large dataset of natural images.

Now, as mentioned earlier, another approach to contextual object recognition is possible. In the next section we discuss such an alternative method based only on interactions between individual object labels in the image.

4.2 Object Based Context (OBC) Model

To provide the necessary analysis of OBC models we pick a representative formulation of CoLA. To stay consistent with the original work, we will use the same notation as in Section 3.3.

At a high level, this representation is built on considering multiple stable segmentations for the input image, resulting in a large collection of segments, though variants also exist using, for example, random segmentations or bounding boxes. Each segment is considered as an individual image and is used as input into a Bag of Features (BoF) model for recognition. Each segment is assigned a list of candidate labels, ordered by confidence. The segments are modeled as nodes of a Conditional Random Field (CRF), where location and object co-occurence constraints are imposed. Finally, based on local appearance and contextual agreement, each segment receives a category label.

4.2.1 Appearance

As the CoLA approach relies on segmentation based recognition, segment appearance is quantified as in Section 3.1. To review, segments are classified based on a simple nearest neighbor rule with the un-normalized distance of the test segment S_q to class c as:

$$d(S_q, c) = \min_i d(S_q, I_{ic}) = \min_i \|\phi(S_q) - \phi(I_{ic})\|_1. \tag{4.9}$$

Segment S_q is assigned to its closest category $c_1(S_q)$:

$$c_1(S_q) = \operatorname*{argmin}_c d(S_q, c). \tag{4.10}$$

Similarly, the S_q is assigned to the rest of the categories:
$c_i(S_q) = \text{sort}(d(S_q, c_i)), \forall\ 1 \leq i \leq n$, with sorting in ascending order of distance. In order to construct a probability distribution over category labels for image query segment, we introduce the following definition:

$$p(c_i|S_q) = \left[1 - \frac{d(S_q, c_i)}{\sum_{j=1}^{n} d(S_q, c_j)}\right], \tag{4.11}$$

and is proportional to the nearest neighbor distance between the query segment S_q and the category: $d(S_q, c)$.

4.2.2 Location and Co-Occurrences

To incorporate a complete notion of visual context, both spatial and semantic (co-occurrence of labels) contexts must be included into the recognition system. A CRF is used to learn the conditional distribution over the class labeling given an image segmentation. Here, the CRF formulation uses a fully connected graph between segment labels instead of a sparse one, which yields a much simpler training problem, since the random field is defined over a relatively small number of segments rather than a huge number of raw pixels or small patches.

Context Model. Given an image I, its corresponding segments $S_1, \ldots, S_k$, and probabilistic per-segment labels $p(c_i|S_q)$ (as in [76]), we wish to find segment labels $c_1, \ldots, c_k \in \mathcal{C}$ such that all agree with the segments' content and are in contextual agreement with one other.

This interaction is modeled as a probability distribution:

$$p(c_1 \ldots c_k|S_1 \ldots S_k) = \frac{B(c_1 \ldots c_k) \prod_{i=1}^{k} p(c_i|S_q)}{Z(\phi_0, \ldots \phi_r, S_1 \ldots S_k)}, \tag{4.12}$$

$$\text{with } B(c_1 \ldots c_k) = \exp\Big(\sum_{i,j=1}^{k} \sum_{r=0}^{q} \alpha_r \phi_r(c_i, c_j) \Big), \tag{4.13}$$

where $Z(\cdot)$ is the partition function, q is the number of pairwise spatial relations, and α_r is the weighting for each relation. The marginal terms $p(c|S)$, which are provided by the recognition system, are explicitly separated from the interaction potentials $\phi_r(\cdot)$. To incorporate both semantic and spatial context information into object categorization, namely into the CRF framework, context matrices are constructed.

Location. Spatial context is captured by co-ocurrence matrices for each of the four pairwise relationships (above, below, inside and around). The matrices contain the frequency among objects labels in the four different configurations, as they appear in the training data. An entry (i, j) in matrix $\phi_r(c_i, c_j)$, with $r = 1, \ldots, 4$, counts the number of times an object with label i appears with an object label j for a given relationship r. Figure 3.3 illustrate the counts over the four different relationships for the MSRC dataset.

Co-occurrence Counts. The co-occurrences of category labels is computed directly from the above mentioned spatial co-occurrences matrices as described in Section 3.3. An entry (i, j) in the co-occurrence matrix counts the times an object with label i appears in a training image with an object with label j. The diagonal entries correspond to the frequency of the object in the training set: $\phi_0(c_i, c_j) = \phi'(c_i, c_j) + \sum_{k=1}^{|\mathcal{C}|} \phi'(c_i, c_k)$, where $\phi'(\cdot) = \sum_{r=1}^{q} \phi_r(c_i, c_j)$.

Therefore the probability of some labeling is given by the model: $p(l_1 \dots l_{|\mathcal{C}|}) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i,j\in\mathcal{C}} \sum_{r=0}^{q} l_i l_j \cdot \alpha_r \cdot \phi_r(c_i, c_j)\Big)$, with l_i indicating the presence or absence of label i. For a detailed description of this example OBC model, refer to Chapter 3.

4.3 SBC vs. OBC: a Comparison

In the previous section we formulated both the SBC and the OBC models in a manner suitable for a direct comparison. In the following section we show that both definitions of context extract the same physical and semantic information from images and training set, yet use it quite differently.

4.3.1 Differences and Similarities

Let us compare

$$p(O_n|\mathbf{v}) = p(\mathbf{v}_L|O_n, \mathbf{v}_C)p(o_n|\sigma, x)\sum_{i=1}^{k} p(o_n|C_i)p(C_i|\mathbf{v}_C) \tag{4.14}$$

to

$$p(c_1 \dots c_k|S_1 \dots S_k) = \frac{B(c_1 \dots c_k)\prod_{i=1}^{k} p(c_i|S_q)}{Z(\phi_0 \dots \phi_r, S_1 \dots S_k)} \tag{4.15}$$

term by term.

Spatial Context:

$$p(\mathbf{v}_L|O_n, \mathbf{v}_C) \leftrightarrow \frac{\exp\Big(\sum_{i,j=1}^{k}\sum_{r=1}^{q} \alpha_r \phi_r(c_i, c_j)\Big)}{Z(\phi_1 \dots \phi_r, S_1 \dots S_k)}, \tag{4.16}$$

where $p(\mathbf{v}_L|O_n, \mathbf{v}_C)$ refers to estimating the probability of the local patch $\mathbf{v}_L$ containing the object of interest O_n, given the scene information $\mathbf{v}_C$. In other words, assuming the scene context and object identity, where are the probable locations for the object of interest? Similarly, $\frac{\exp(\sum_{i,j=1}^{k}\sum_{r=1}^{q} \alpha_r \phi_r(c_i,c_j))}{Z(\phi,S_1 \dots S_k)}$, the spatial component of $\frac{B(c_1 \dots c_k)}{Z(\phi,S_1 \dots S_k)}$, estimates approximately the same information. Given all the

potential objects in the scene, the probability of each spatial arrangement of objects is calculated. However, instead of estimating the absolute location for each candidate object individually, the relative pairwise locations of all objects are chosen simultaneously.

Appearance:

$$p(o_n|\sigma, x) \leftrightarrow p(c_i|S_q), \tag{4.17}$$

where $p(o_n|\sigma, x)$ is the likelihood of a particular object being present in a given region of the image (region is defined by scale and location). In turn, $p(c_i|S_q)$ is also the likelihood of a particular object, c_i being present at a particular region of the image, yet here the region is defined by segment S_q.

Semantic (co-occurrence) Context:

$$\sum_{i=1}^{k} p(o_n|C_i)p(C_i|\mathbf{v}_C) \leftrightarrow \frac{\exp\Big(\sum_{i,j=1}^{k} \alpha_0\phi_0(c_i, c_j)\Big)}{Z(\phi_0, S_1 \ldots S_k)}. \tag{4.18}$$

Here, $\sum_{i=1}^{k} p(o_n|C_i)p(C_i|\mathbf{v}_C)$ captures the semantic context via the scene information C_i. Once the scene category $p(C_i|\mathbf{v}_C)$ is estimated, the most probable object label, o_n, is chosen from the potential labels in the given scene. Alternatively, $\frac{\exp(\sum_{i,j=1}^{k} \alpha_0\phi_0(c_i,c_j))}{Z(\phi_0,S_1 \ldots S_k)}$, provides a likelihood of all possible combinations of objects that the existing segments, $S_1 \ldots S_k$, may be labeled with. Only pairwise relationships between object co-occurrences are learned during training.

As shown above, the SBC and OBC models are analogous in terms of the information and statistics they use to apply contextual reasoning to object recognition. However, as we show next, there are a number of differences between the two models that make the OBC model more attractive and empirically more effective.

4.3.2 Inference

In estimating quantities 4.14 and 4.15, it is crucial to understand the processes of inferring the likelihoods, thresholding, and error propagation. In the case

of Gist, one first estimates the scene context $p(C_i|\mathbf{v}_C)$, and subsequently the object label, given the chosen scene $p(o_n|C_i)$, as illustrated in Figure 4.2(a). In particular, choosing the scene context is critical since it constrains the possible object labels in the image. Inferring an incorrect scene from the context reduces the likelihood of identifying the true object labels, see Figure 4.4 (3 bottom rows in column (b)). Furthermore, only the scenes that have been predefined or learned in training may be considered for an input image, however, objects that exist in the training set may appear in different configurations (scenes) from those in test images. Thus, the accuracy of identifying the labels of objects that exist in an image is critically dependent on identifying the correct scene label for the image. In turn, scene information also requires learning, and is heavily dependent on the training set or manually defined rules.

Alternatively CoLA, an OBC model, Figure 4.2(b), employs a simple representation and an efficient algorithm for extracting information from visual input without committing to a scene label in a preprocessing stage. Using the traditional Bayesian likelihood estimation of a particular image region being a given object, $p(c_i|S_q)$, a graphical model selects the particular object labels based on the object category co-occurence and spatial relations according to the training data.

Although scene based context is not required for accurate object recognition with an OBC model, we think that scene-level information is indeed an interesting notion. Using the CoLA formulation, this information can available as a byproduct, rather than as an input, as in Gist. Once the probability of a given set of object labels, $\frac{B(c_1 \ldots c_k)}{Z(\phi_0 \ldots \phi_r, S_1 \ldots S_k)}$, is determined, that set of labels can be mapped to a particular scene.

4.3.3 Training

Training is a crucial part of any classification task, and object recognition in particular. The two key aspects pertaining to training data are the level of detail in

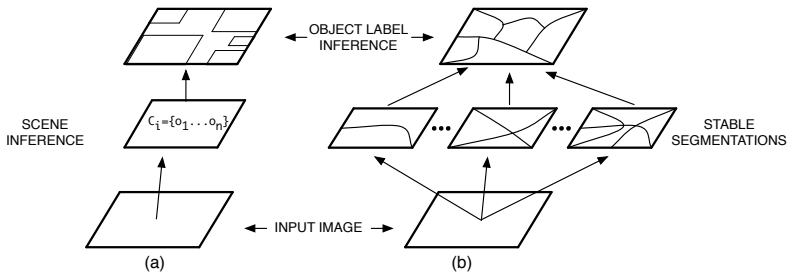


Figure 4.2: Inference with Gist (a) and CoLA (b). Inferring the object labels using Gist requires one first to commit to a scene category and only then infer the object label; with CoLA, no such commitment is necessary.

the data labeling and the training set size. Scene based approaches require a large training set since many examples are needed to capture not only the statistics of the object category, but also its scene context [81, 93]. Furthermore, training data must be labeled with the individual object labels, and also with the scene labels. To our knowledge, the majority of object recognition datasets do not contain scene definitions and moreover, it is not clear how to define the scene context. For example, nearly identical scenes may be identified as either *beach* or *coast*, or even as *shore*. Potentially, word hierarchies such as *WordNet* may be used to resolve such ambiguities, but this adds another layer of complexity to the model. Also, as the number of object categories increases, the number of scenes will likely also increase as well and ambiguities between scenes will also be greater.

Approaches based on individual object interactions, however, require considerably less training data as only object appearance and object co-occurrence needs to be learned. In [23, 76] only 30 examples per category were used for training. Only object labels themselves are necessary for training, rather than scene context.

4.3.4 Scalability

One drawback of the OBC model, is that the required example interactions between object labels are rather sparse in the currently available datasets, (see

Figure 3.3) . Not many object categories co-occur in the same images. However, with the inclusion of many more object categories, the contextual matrices will only get richer and importance of contextual constraints will be even more evident. Note that the complexity of learning co-occurrences is only quadratic in the number of categories since only pairwise relations are computed.

The approach of Gist type methods, which heavily rely on scene information, will perhaps only suffer from an inclusion of additional object categories. New scenes will have to be defined, and the problem of scene inference given the semantic context, $\mathbf{v}_C$ will become even more ambiguous.

4.4 Empirical Comparison of Contextual Models

In this section we perform an empirical comparison of the two discussed approaches. We used the same subset of the LabelMe, [80], dataset for the experimental comparison as was done by [81]. We trained and tested the CoLA approach with twelve categories. The training set has 15691 images and 105034 annotations and the test set has 560 images and 2026 annotations. The test set comprises images of street scenes and indoor office scenes. To avoid overfitting, street scene images in testing were photographed in a different city from the images in the training set. Figure 4.4 shows localization and recognition accuracy for example images taken from the LabelMe dataset using Gist and CoLA. Column (c) in Figure 4.4 shows the accuracy of localization using the stable segmentations used by CoLA. Since this database contains many more categories than just twelve that were chosen by [81], some of the localized regions are not labeled, due to low recognition accuracy, to avoid a forced choice label. In this experiment we mark regions as 'unknown' if the maximum label probability is less than or equal to chance. (On average, of 54 segments per image, 1.51 were labeled as 'unknown'.) Note that the segmentation based approach not only eschews the step of predicting the scene first, thus avoiding as possibly incorrect retrieval set, but it also provides

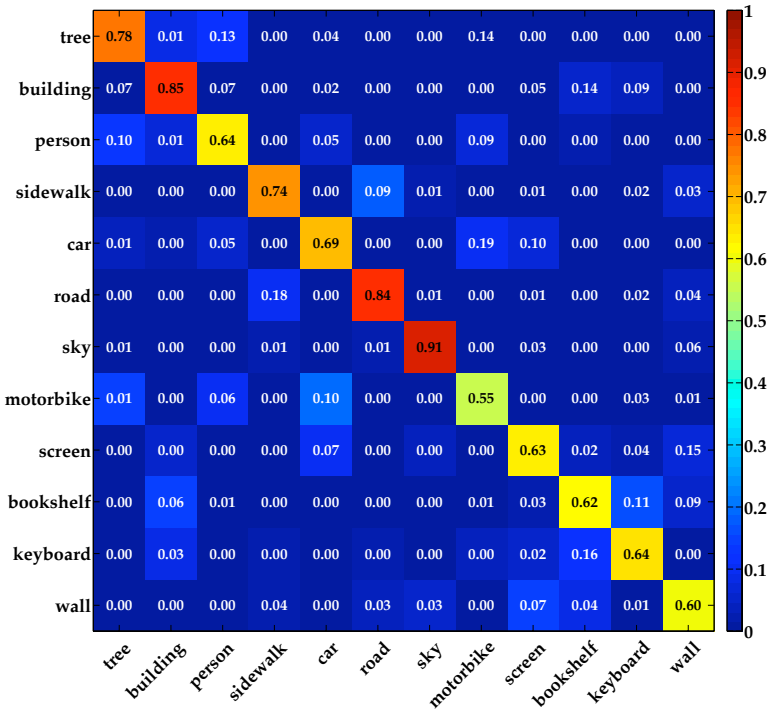


Figure 4.3: Confusion Matrix for the LabelMe dataset using CoLA.

accurate localization with object boundaries rather than bounding boxes. We refer the reader to [81] and [23] for implementation details and runtime complexity for both Gist and CoLA.

The results in Figure 4.4 show qualitative differences between the two compared models; however, we wish to evaluate the models quantitatively. In Table 4.4, we report recognition accuracy, true positive rate (TPR), and the false positive rate (FPR) for both models. The results for Gist were taken directly from ROC curves in [81]; results for CoLA are computed from the confusion matrix shown in Figure 4.3[1].

Since [81] formulated the recognition problem as a detection task, they emphasized the low FPR per bounding box per category, while in recognition problems the TPR is maximized with less attention to FPR. We show TPR rates for the FPR suggested in [81], and show corresponding FPR per image per category,

[1]TPR corresponds to the diagonal entries of the confusion matrix and the FPR is the hollow confusion matrix column sum; both refer to the confusion matrix in Figure 4.3

Table 4.1: Recognition accuracy (true positive rate TPR) and false positive rate (FPR) per image per category for both Gist and CoLA approaches. **Gist (low FPR):** TPR for the FPR per image per category that was suggested in [81]. **Gist (high TPR):** FPR (from ROC curves in [81]) per image per category for TPR that is comparable to that of CoLA. **SVM (no context):** FPR (also from [81]) per image per category for TPR, without aid of context, that is comparable to one achieved by CoLA. **CoLA:** TPR and FPR per image per category using CoLA. Note that TPR for CoLA is almost 3 fold greater than for Gist (**70.9%** vs. **27.2%**), while FPR for CoLA is almost two orders of magnitude lower than that of Gist (**0.02** vs. **1.14**) per image per category.

	Gist (low FPR)		Gist (high TPR)		SVM (no context)		CoLA	
category	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
tree	9.59%	1.05	76.0%	36.1	53.1%	41.9	78.1%	0.03
building	7.29%	2.09	85.3%	108	60.2%	111	85.8%	0.04
person	21.1%	0.78	68.5%	24.8	78.4%	25.1	64.0%	0.02
sidewalk	7.98%	2.11	70.2%	52.6	66.0%	54.5	74.4%	0.02
car	68.0%	0.03	68.6%	0.83	44.4%	0.89	69.6%	0.03
road	37.0%	0.86	84.6%	31.6	64.3%	29.7	84.7%	0.03
sky	34.5%	1.49	89.6%	106	60.1%	107	91.9%	0.01
motorbike	48.6%	0.81	55.6%	1.19	63.9%	2.10	55.4%	0.02
screen	50.0%	1.17	64.2%	3.81	88.3%	4.57	68.1%	0.02
bookshelf	13.0%	1.04	61.7%	17.9	46.8%	27.8	59.1%	0.03
keyboard	26.5%	0.61	62.0%	10.3	81.4%	15.2	64.5%	0.01
wall	3.08%	0.88	47.7%	84.6	29.2%	61.7	60.0%	0.02
mean	**27.2%**	**1.14**	**63.2%**	**39.9**	**61.4%**	**40.2**	**70.9%**	**0.02**

shown in hypercolumn "Gist (low FPR)", rather than per bounding box. TPR and FPR, per image per category, for CoLA are shown in hypercolumn "CoLA". Note that TPR for CoLA is almost 3 fold greater than for Gist, while FPR for CoLA is almost two orders of magnitude lower than that of Gist. This comparison, however, does not isolate the effectiveness of the contextual model itself. In the case of Gist, the underlying detector or classifier (SVM) may be weak, or in the case of CoLA the stable segmentations may be useless. Similar to the work of [76], where the authors show the significant improvement yielded by including of context in the recognition framework (see Table 3.2.2), we evaluate the relative improvement of adding context to the Gist method. In Table 4.4, we show the TPR (at competitive

rates) and FPR for the Gist approach with context "Gist (high TPR)", and only the SVM detector module of the "Gist (SVM no context)". Means of both TPR and FPR are within one standard deviation of each other, and the difference between them is not statistically significant. This suggests that recognition rates of the full Gist approach is hindered by its contextual model rather than the underlying detector or classifier. A possible avenue for improvement of the Gist approach could be to entertain multiple scene category hypotheses, rather than committing to the most probable one.

Figure 4.4: Recognition results for example from LabelMe dataset. (a) Original image. (b) Detected objects by Gist. (c) Recognized objects by CoLA. (d) Ground truth object labeling.*Best viewed in color.*

Chapter 5.

Conclusion

Object recognition has been of great interest to scientists across many fields, from psychology to computer vision. With the increasing computing resources in the last decade, many algorithms for automatic object recognition have been proposed. In the earlier models to automate object recognition, researchers in computer vision attempted to utilize image segmentation as a means of partitioning the images into a set of regions for one to one correspondence with the actual objects in the scene. However, unable to produce viable segmentations, this approach was abandoned at large, and approaches based of scanning windows were adopted. Often segmentation has not been used in recognition due of the difficulty of obtaining segments corresponding to the objects of interest. However, in this work we solve this problem by relying on a shortlist of potentially meaningful segmentations (identified by a stability criterion), which significantly increase the chance of extracting suitable segments. Incorporating this segmentation method with a simple BoF model, we showed recognition accuracy at the level comparable with the state-of-the-art (Table 3.2.2, [104]).

More importantly, image segmentation can not only aid in achieving competitive recognition rates, it in fact improves object recognition and categorization by adding accurate object localization and multi-class categorization capabilities

to an off-the-shelf categorization system, as was shown on CALTECH and PASCAL datasets. Representing image regions as segments, rather than bounding rectangles, offers features otherwise unavailable: object shapes, increased signal to noise ratio, and other valuable statistics. Also, on an slightly orthogonal note, the proposed approach of segmenting test images and recognizing individual segments, provides an intuitive framework for semantic context based object categorization.

Over the past few years, the role of contextual models has become more prominent in object recognition systems. As the field of contextual object recognition in computer vision evolves, SBC and OBC models have emerged. In the approach proposed by [91], an example of SBC model, contextual information is captured by the statistical summary of the image. This approach may be related to the contextual processing in the human visual system. The SBC model is very intuitive and potentially efficient. Yet, an alternative, OBC based, formulation of context for recognition has recently been proposed. With the OBC model, a relationship between individual objects is induced, instead of capturing the context of the scene by its low level holistic representation.

In this work we have compared the two contextual models for object recognition and showed similarities and differences between them. In particular both models capture analogous physical and semantic information from the image. Yet, we demonstrated analytically that the OBC model, although computationally more expensive due to the cost involved in computing the stable segmentations, gives rise to a simpler inference problem. Using the LabelMe database, we empirically compared the two models and showed that CoLA, an approach using an OBC model, considerably outperformed Gist, a SBC based method. The two major differences between OBC and SBC models are the use of stable segmentations vs. sliding windows; and the notion of context: object based vs. scene based.

The significant improvement in performance of the OBC model is due in part to the use of the stable segmentations. In particular, multiple stable segmentations are able to represent the image in a compact and informative manner

for the task of object recognition. Without such a compressed representation of image partitions, it is combinatorially difficult to enforce contextual constraints between individual objects. Thus, many algorithms tend to settle for scene based contextual connections, which in turn lead to rather confined and weak contextual support. With only a single segmentation it is virtually impossible to identify all the objects or their parts to perform recognition. On the other hand, considering *all* possible segmentations would greatly hinder the false positive rates of the recognition system, as suggested by the experiment using thousands of bounding boxes. Thus, a compact representation of multiple stable segmentations facilitates the construction of object recognition models with low false positive rates of recognition and contextual constraints strong enough to correct and disambiguate appearance based pitfalls. We believe that the shortlist of stable segmentations (aiming for only those segmentations that matter) is the essential substrate for competitive Object Based Context models for object recognition and categorization.

References

[1] E. Aminoff, N. Gronau, and M. Bar. The Parahippocampal Cortex Mediates Spatial and Nonspatial Associations. *Cerebral Cortex*, 17(7):1493, 2007.

[2] M. Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8), 2004.

[3] M. Bar and E. Aminoff. Cortical Analysis of Visual Context. *Neuron*, 38(2):347–358, 2003.

[4] 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. *CVPR*, 2003.

[5] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[6] F. Benezit, T. Cour, and J. Shi. Spectral segmentation with multi-scale graph decomposition. In *CVPR*, 2005.

[7] I. Biederman, RJ Mezzanotte, and JC Rabinowitz. Scene perception: detecting and judging objects undergoing relational violations. *Cognitve Psychology*, 14(2):143–77, 1982.

[8] Irving Biederman, Robert J. Mezzanotte, and Jan C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143–177, April 1982.

[9] C. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, 1995.

[10] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *European Conference on Computer Vision*, May 2002.

[11] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. *ECCV*, pages 350–362, 2004.

[12] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multi-scale graph decomposition. In *CVPR*, 2005.

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

[14] J.L. Davenport and M.C. Potter. Scene Consistency in Object and Background Perception. *Psychological Science*, 15(8):559–564, 2004.

[15] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002.

[16] M. Everingham et al. The 2005 pascal visual object classes challenge. In *In Proc. of PASCAL Challenge Workshop, LNAI,*, 2006.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge (VOC) Results, 2007.

[18] Christiane D. Fellbaum. *WordNet : An Electronic Lexical Database*. MIT Press, 1998.

[19] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *CVPR*, 2003.

[20] Michael Fink and Pietro Perona. Mutual boosting for contextual inference. *Advances in Neural Information Processing Systems 16*, 2004.

[21] M.A. Fischler and T.M. Strat. Recognizing objects in a natural environment: a contextual vision system (CVS). *Proceedings of a workshop on Image understanding workshop table of contents*, pages 774–796, 1989.

[22] A. Friedman. Framing pictures: the role of knowledge in automatized encoding and memory for gist. *J Exp Psychol Gen*, 108(3):316–55, 1979.

[23] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using Co-Occurrence, Location and Appearance. In *CVPR*, 2008.

[24] Zoubin Ghahramani and K. A Heller. Bayesian sets. In *NIPS*, 2005.

[25] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[26] R.D. Gordon. Attentional Allocation During the Perception of Scenes. *Journal of Experimental Psychology Human Perception and Performance*, 30(4):760–77717, 2004.

[27] N. Gronau, M. Neta, and M. Bar. Integrated Contextual Representation for Objects Identities and Their Locations. *Journal of Cognitive Neuroscience*, 20(3):1–18, 2008.

[28] A.R. Hanson and E.M. Riseman. Visions: A computer vision system for interpreting scenes. *Computer Vision Systems*, pages 303–334, 1978.

[29] R.M. Haralick. Decision making in context. *PAMI*, 5(4):417–428, July 1983.

[30] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[31] Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. *CVPR*, pages 695–702, 2004.

[32] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238, 1995.

[33] D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. *CVPR*, 2006.

[34] A. Hollingworth and J.M. Henderson. Does consistent scene context facilitate object perception. *Journal of Experimental Psychology: General*, 127(4):398–415, 1998.

[35] H.C. Hughes, G. Nozawa, and F. Kittler. Global precedence, spatial frequence channels, and the statistics of naturals scenes. *J. of Cog. Neuroscience*, 8(3):197–230, May 1996.

[36] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[37] D.R.T. Keeble, F.A.A. Kingdom, and M.J. Morgan. The orientational resolution of human texture perception. *Vision Research*, 37(21):2993–3007, 1997.

[38] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[39] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, 2:83–97, 1955.

[40] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. volume 2, pages 1150–1157, 2003.

[41] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1284–1291, 2005.

[42] C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. *Proc. of CVPR*, 2008.

[43] M.S. Landy and I. Oruc. Properties of second-order spatial frequency channels. *Vision Research*, 42:2311–2329, 2002.

[44] T. Lange, V. Roth, M.L. Braun, and J.M. Buhmann. Stability-based validation of clustering solutions. In *NIPS*, 2002.

[45] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proc. CVPR*, 2:2169–2178, 2006.

[46] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[47] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[48] A. Levin and Y. Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. *ECCV*, 2006.

[49] E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13, 2001.

[50] T. Lindeberg. Principles for automatic scale selection. Technical Report ISRN KTH/NA/P-98/14-SE, Royal Inst. of Tech., 1998.

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

[52] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue integration in image segmentation. In *International Journal on Computer Vision (ICCV)*, 1999.

[53] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, September 2007.

[54] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, May 2004.

[55] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[56] M. Meilă and J. Shi. Learning segmentation with random walk. In *Proc. of NIPS*, pages 873–879, 2001.

[57] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Multiple object class detection with a generative model. *CVPR*, 2006.

[58] JW Modestino and J. Zhang. A Markov random field model-based approach to image interpretation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(6), 1992.

[59] G. Mori, X. Ren, AA Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *CVPR*, 2004.

[60] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the tree: a graphical model relating features, objects and the scenes. *NIPS.*, 2003.

[61] D. Navon. Forest before trees: The precedence of global features in visual perception. *Congitive Psychology*, 1977.

[62] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[63] M.E. Nilsback, A. Zisserman, M. Vision, and M.P. Kumar. A visual vocabulary for flower classification. *CVPR*, 2006.

[64] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *CVPR*, 2006.

[65] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-Features Image Classification. *LNCS*, 2006.

[66] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[67] A. Oliva and A. Torralba. The role of context in object recognition. *TRENDS in Cognitive Science*, 2007.

[68] P. Paatero and U. Tapper. Least squares formulation of robust non-negative factor analysis. *CILS*, 37:23–35, 1997.

[69] S. E. Palmer. *Vision Science*. MIT Press, 1999.

[70] S.E. Palmer. The effects of contextual scenes on the identification of objects. *Memory and Cognition*, 3(5):519–526, 1975.

[71] T. Poggio, E. Gamble, and J. Little. Parallel integration of vision modules. *Science*, 242:436–440, 1988.

[72] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–71, 2000.

[73] John Prager, Jennifer Chu-Carroll, and Krzysztof Czuba. Question answering using constraint satisfaction: QA-by-dossier-with-constraints. *ACL*, 2004.

[74] A. Rabinovich, T. Lange, J. Buhmann, and S. Belongie. Model order selection and cue combination for image segmentation. In *CVPR*, 2006.

[75] A. Rabinovich, A. Vedaldi, and S. Belongie. Does image segmentation improve object categorization? *UCSD Technical Report cs2007-0908*, 2007.

[76] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *International Conference on Computer Vision*, 2007.

[77] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 2005.

[78] V. Roth and B. Ommer. Exploiting low-level image segmentation for object recognition. In *DAGM*, 2006.

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

[80] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. LabelMe: a database and web-based tool for image annotation. *MIT AI Lab Memo AIM-2005-025*, 1:1–10, 2005.

[81] Bryan C. Russell, Antonio Torralba, Ce Liu, Rob Fergus, and William T. Freeman. Object recognition by scene alignment. In *NIPS*, 2007.

[82] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. *JNLPBA*, 2004.

[83] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML*, 2005.

[84] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[85] 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.

[86] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *IJCV*, 2007.

[87] Amit Singhal, Jiebo Luo, and Weiyu Zhu. Probabilistic spatial context models for scene content understanding. *CVPR*, 01:235, 2003.

[88] T.M. Strat. *Natural object recognition*. 1992.

[89] T.M. Strat and M.A. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *Pattern Analysis and Machine Vision*, 13(10):1050–1065, October 1991.

[90] R. Tibshirani, Walther G., and T. Hastie. Estimating the number of clusters via the gap statistic. *Journal of Royal Statistical Society B*, 63(2):411–423, 2001.

[91] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision (IJCV), 53(2):153–167, 2003.*, 2003.

[92] A. Torralba. Modeling global scene factors in attention. *JOSA*, 2003.

[93] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, MIT, 2007.

[94] A. Torralba, A. Oliva, M.S. Castelhano, and J.M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[95] Z.W. Tu and S.C. Zhu. Image segmentation by data driven markov chain monte carlo. In *ICCV*, pages 131–138, 2001.

[96] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek. Evaluation of color descriptors for object and scene recognition. *IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska*, 2008.

[97] A. Vedaldi, 2006. http://vision.ucla.edu/ vedaldi/code/bag/bag.html.

[98] Jakob Verbeek and Bill Triggs. Scene segmentation with crfs learned from partially labeled images. In *NIPS*, 2007.

[99] P. Viola and M. Jones. Robust real time object detection. *IJCV*, 2002.

[100] Y. Weiss. Segmentation using eigenvectors: a unifying view. pages 975–982, 1999.

[101] M. Welling and M. Weber. Positive tensor factorization. *Pattern Recogn. Lett.*, 22(12):1255–1261, 2001.

[102] Lior Wolf and Stanley Bileschi. A critical view of context. *International Journal of Computer Vision (IJCV)*, 2006.

[103] S.X. Yu and J. Shi. Object-specific figure-ground segregation. *CVPR*, 2003.

[104] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.