Modeling Context for Image Understanding:

When, For What, and How?

Devi Parikh

Department of Electrical and Computer Engineering,

Carnegie Mellon University

A thesis submitted for the degree of

Doctor of Philosophy

April 3, 2009

Thesis Committee

Prof. Tsuhan Chen
Department of Electrical and Computer Engineering,
Cornell University
Department of Electrical and Computer Engineering,
Carnegie Mellon University

Dr. Larry Zitnick
Microsoft Research, Redmond

Dr. Rahul Sukthankar
Intel Research, Pittsburgh
Robotics Institute,
Carnegie Mellon University

Prof. Vijayakumar Bhagavatula
Department of Electrical and Computer Engineering,
Carnegie Mellon University

Prof. Martial Hebert
Robotics Institute,
Carnegie Mellon University

I dedicate this thesis to my parents:

I couldn't have asked for more.

Acknowledgements

I would like to thank my advisor, Prof. Tsuhan Chen. Thank you for your support and guidance through out the four years, and for your optimism and confidence in me every step along the way. I would also like to thank my thesis committee: Dr. Larry Zitnick (it has been a lot of fun and a great learning and productive experience working with you), Dr. Rahul Sukthankar (thank you for taking the time and effort to go out of your way to guide and advise me anytime I needed it, even though you didn't have to), and Prof. Vijayakumar Bhagavatula and Prof. Martial Hebert (thank you for your input on this thesis).

I am grateful for the intellectually stimulating environment at Carnegie Mellon. I have benefited immensely from the seminars and talks and reading groups that I attended almost daily. The unending discussions and debates with my floor-mates on topics ranging from details of a mathematical concept to philosophies on which research directions are more noble than others have made grad school years memorable in more than one ways.

Abstract

A key problem in computer vision is image understanding, which we define as the task of recognizing every object/region in the scene. Traditionally, this has been accomplished by considering the information within each object/region to be recognized. Incorporating contextual information, i.e. information other than the appearance information of the object, for image understanding has received significant attention in recent works. Contextual information is often learnt in a supervised manner and utilized to enhance performance of higher level tasks such as object recognition or detection. In this thesis, we take a closer look at the role of context in image understanding. Specifically, we ask three questions. First: *When* is context really helpful? We show, through computer vision experiments as well as human studies, that context provides improvements in recognition performances only when the appearance information is weak (such as in low resolution images or in the presence of occlusion). Second: *For what* tasks can contextual information be leveraged? We show that apart from high-level tasks of recognition and detection, contextual information can be effectively leveraged for low level tasks as well, such as identifying salient or representative patches in an image. Lastly, *How* can context be learnt? Or alternatively, how much contextual information can be extracted in an unsupervised manner? We propose a unified hierarchical representation for contextual interactions or spatial patterns among visual entities at all levels, from low-level features to parts of objects, objects, groups of objects and ultimately the entire scene. We present results of our approach on a variety of datasets such as object categories, street scenes and natural scene images.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image understanding can be thought of as the task of associating every object and region visually depicted in an image of a scene, and perhaps the entire scene itself, to a semantic concept humans understand, as illustrated in Figure 1.1. This task of image understanding manifests itself through several popular computer vision tasks such as object recognition, object detection and scene recognition. A traditional machinery employed for these tasks is to collect a set of labeled images as training data, extract features like color, shape and texture that describe the appearance of the objects and train classifiers that hold models of these different object types. For instance, a model can be learnt that indicates that cows are brown, grass is green, airplanes have a smooth texture, and human faces are oval. A new image region or object is matched against these models, and is classified as the object type that best matches the region.

Recent works have observed that the cues of an object's type are not present only within the object boundaries. The visual information surrounding an object also holds strong cues about the identity of an object. For example, cows are often

Figure 1.1: Image understanding as an image labeling task

present on grass and cats do not fly in the sky. Hence, if an object is present in the sky, it is needless to consider the possibility of it being a cat. Many works have attempted to incorporate this surrounding contextual information, as opposed to the appearance information of the objects alone, into the image understanding pipeline for increased efficiency as well as accuracy.

A subtle connection can be made to previous works that classify entire images for the presence of an object, where the classifiers may inadvertently look for green grass to detect the presence of a cow. This was, interestingly, often viewed as a downside of these approaches, and hence works started focusing on the object localization task. However, more recently, the value in contextual information has been (re)discovered, and is being incorporated in a more explicit and principled manner.

While significant progress has been made in incorporating the contextual information and understanding it's impact on enhanced image understanding, we believe there are several aspects of the role of context in image understanding that have been largely ignored. These are the aspects explored in this thesis. For instance, context has mostly been exploited as a post-processing step to incorporate obvious semantics. We study the scenarios under which context is really necessary, and most beneficial over the appearance information. Context has

mostly been explored for the high level task of object recognition and detection. We explore the role of context for the low-level task of picking salient or representative patches from images, and in-turn evaluate the effectiveness of these selected patches for the task of image classification. And finally, most works employ the contextual information through supervised methods where labeled training data is used to learn these contextual relationships. We investigate how much of this contextual information can be learnt in an unsupervised way. We use a hierarchical representation to describe the contextual relationships among low-level as well as high-level visual entities in images.

The rest of this thesis is organized as follows. Chapter 2 describes our work on studying *When* contextual information is necessary. Chapter 3 presents our work on exploring *For What* tasks contextual information can be leveraged. Chapters 4 and 5 present our approach on *How* the contextual information, represented hierarchically, can be learnt in an unsupervised way. We provide an introduction and relevant background for each of these three questions in their relevant chapters. The thesis is concluded in Chapter 6 with a discussion of potential future work.

Chapter 2

When: Recognition in Small Images

Summary

Traditionally, object recognition is performed based solely on the appearance of the object. However, relevant information also exists in the scene surrounding the object. As supported by our human studies, this contextual information is *necessary* for accurate recognition in low resolution images. The same can not be said about images of high resolution. Thus, this scenario with impoverished appearance information, as opposed to using images of higher resolution, provides an appropriate venue for studying the role of context in recognition.

In this chapter, we explore the role of context for dense scene labeling in small images. Given a segmentation of an image, our algorithm assigns each segment to an object category based on the segment's appearance and contextual information. We explicitly model context between object categories through the use

of relative location and relative scale, in addition to co-occurrence. We perform recognition tests on low and high resolution images, which vary significantly in the amount of appearance information present, using just the object appearance information, the combination of appearance and context, as well as just context without object appearance information (blind recognition). We also perform these tests in human studies and analyze our findings to reveal interesting patterns. We find that contextual information increases recognition accuracies only in low-resolution images, where the appearance information is weak. With the use of our context model, our algorithm achieves state-of-the-art performance on MSRC and Corel datasets.

2.1 Introduction

Traditionally, research on recognizing object categories in images has focussed on appearance information pertaining only to the object itself. For instance, parts-based approaches [3, 4] recognize objects by localizing a set of parts corresponding to the local appearance and structure of the object. Popular datasets such as the Caltech datasets [5, 6] have been constructed specifically for such a treatment, where the object to be recognized is found in the center and occupies a significant portion of the image.

In natural images, relevant contextual information about the object also lies in the scene surrounding the object. Recently, many works [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19] have attempted to move beyond a purely appearance-based approach by incorporating context using several approaches. Global scene information, such as global texture [10, 19] or 3D scene information [8], can be

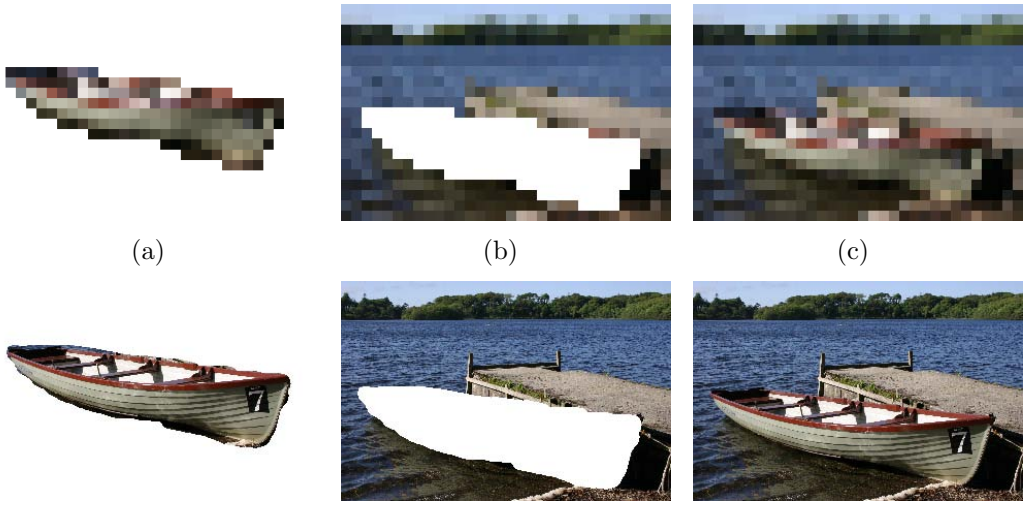


Figure 2.1: Example of recognition using appearance alone (a,d), using context alone, i.e. blind recognition (b, e) and context and appearance combined (c, f) for low resolution images (a, b, c) and high resolution images (d, e, f). For low resolution images, context is *necessary* for recognition given the small amount of information provided by the appearance, which is not the case for high resolution. Hence, we advocate exploring context in low resolution images.

used as context to reduce the set of possible objects that may be present in the scene, or to reduce the possible locations of the objects [10, 11, 8, 18, 19]. Context may also be modeled locally by examining neighboring textures [13, 15], by extracting multi-scale features [12], or by modelling interactions between neighboring regions in the images [12, 14, 16].

Instead of using context to model scene or local texture properties, context may also be used to model higher-level, potentially semantic, interactions among objects [7, 9]. Torralba *et al.* [9] detect easier to recognize objects first, which in turn aid in the detection of harder objects. Hoiem *et al.* [8] use information from multiple object types by taking advantage of viewpoint information about

the scene. Rabinovich *et al.* [7] and Singhal *et al.* [17] proposed the explicit modeling of inter-object context using object co-occurrence, and hand-coded spatial relationships respectively.

There exist several scenarios, as shown in Figure 2.2 in which an object's appearance alone is clearly insufficient for recognition. For instance, the amount of appearance information may be limited due to bad image quality, viewing of a scene from a distance, low image resolution, occlusion, etc. An example is shown in Figure 2.2 (left), where without the context of the rest of the scene (top), it would be hard to recognize the keyboard (bottom). If the amount of intra-class appearance variation is high, or the inter-class appearance variation is low, context may be needed to disambiguate an object's category. For example, as shown in Figure 2.2 (center), clothing varies drastically in appearance and is mainly defined by its position relative to the body. And as seen in Figure 2.2 (right), some object categories such as sky and water, or TV screen and computer monitor have very similar appearance, and may only vary in their relative locations and object surroundings. However, in many scenarios addressed by prior works context is used to increase recognition accuracy in scenarios where the appearance information is clear, and the object categories are visually distinct. In such cases, it is unclear whether improved use of appearance information could give similar performance boosts, and the use of context may be unnecessary.

In this chapter, we explore object level context in the scenario of impoverished image data. Specifically, our goal is dense object labeling in extremely low resolution images. The need for effective computer vision in low resolution images has many practical standings. Low resolution images are space efficient and allow for much faster processing and streaming. Many devices such as cell phone

Figure 2.2: Illustration of a few scenarios where contextual information is necessary for effective recognition. Left: Impoverished appearance information makes it hard to recognize the keyboard in the image without contextual information; Center: diverse appearance information for the category *clothes* makes it difficult to build a consistent appearance model to describe it; Right: Appearance information is similar for two semantically distinct categories of *TV screen* and *computer monitor* thus requiring contextual information to disambiguate.

cameras and web cameras often produce low quality and low resolution images. Images of far away scenes, or images of cluttered complex scenes result in the effective resolution of the individual objects being quite small. The use of low resolution images has also been explored by Torralba *et al.* [21] for the recognition of scene categories and object detection using a large database of labeled images. They find that humans can recognize objects in very low resolution images, even when the objects contain just a few pixels. They hypothesized that humans leverage contextual information to do so, since the appearance information has negligible information. We study this formally to tease apart the contributions of appearance and contextual information for the recognition task for humans and machines. Efros *et al.* [22] recognize human actions in distant videos where the effective resolution of sportsmen is very small.

As we show in later sections, human studies verify that appearance informa-

tion alone is not enough for accurate object recognition in low resolution images. However with the use of context, we find that humans can recognize objects quite reliably, as also observed by Torralba *et al.* [21]. In fact, for the task of blind recognition where appearance information is withheld and only contextual information is given to the subject, recognition accuracy is roughly equal to that of using appearance alone. These studies verify that the task of recognition in low resolution images is an interesting venue for modeling context.

We achieve dense object labeling by assigning labels to a set of pre-computed segments. The segment labels are assigned to be consistent with the contextual information learned from the training data set. The beliefs in a segment's labels are computing using a fully connected Conditional Random Field (CRF) with the segments acting as nodes. Context is modeled using the pairwise potentials of the CRF. This formulation allows for the use of a wide variety of contextual information.

Our contributions in this chapter are as follows. We perform object recognition in low resolution images; an appropriate scenario for exploring context in which context is *necessary* for accurate recognition. We model context explicitly, and incorporate inter-object relationships in terms of relative location and scale in addition to object co-occurrence. To explore the utility of appearance and contextual information we perform tests on both low and high resolution images, using just object appearance information, using context without object appearance (blind recognition), and the combination of appearance and context. These tests were performed both in human and machine experiments. State-of-the-art performances are achieved on the MSRC [23] and Corel [24] datasets.

The rest of the chapter is organized as follows. Section 2.2 describes our con-

text model. Section 2.3 describes the experimental set up for our human studies and machine experiments, and provides results and related analysis.. Section 2.4 raises some interesting points of discussion, followed by a conclusion in Section 2.5.

2.2 Approach

Our goal is to utilize context for recognizing objects in very low resolution images. We obtain these low resolution images by down-sampling images of higher resolution. The aspect ratio of the original image is maintained while reducing the larger dimension to 32 pixels. Torralba *et al.* [21] show that humans can recognize objects in 32×32 images, which our human studies also confirm. Further down-sampling results in a significant degradation in performance [21]. We also apply our method to the original resolution images to study the trade off between appearance and context in different scenarios. The following discussion is common for images of either resolution.

The task we consider is to semantically label every pixel in an image. We approach this task at the region or segment level since good spatial support is shown to significantly help recognition [25, 26]. Hence, our task is to recognize the content of every segment in an image from a pre-determined list of C possible classes. In addition to the appearance information pertaining to the region itself, which we refer to as the data term, we wish to capture the interactions among the different segments through context.

We model this through a fully connected pairwise Conditional Random Field (CRF) similar to [7], where each node corresponds to a segment in the image, and the edges correspond to pair-wise contextual interactions between the segments.

In our experiments, the number of segments per image was on average 7 and never exceeded 17, which made such a model feasible. For more complex scenarios containing a larger number of segments, the structure of the graphical model should be intelligently chosen or learnt from data.

We define the conditional probability of our class labels given the segments within our CRF as

$$P(\mathbf{c}|\mathbf{S}) = \frac{1}{Z}\prod_{i=1}^{N}\Psi_i(c_i)\prod_{i,j=1}^{N}\Phi_{ij}(c_i,c_j), \tag{2.1}$$

where Z is the partition function. The data term $\Psi_i(c_i)$ computes the probability of class c_i given the appearance of segment $S_i \in \{S_1,\ldots,S_N\}$. The pair-wise potentials $\Phi_{ij}(c_i,c_j)$ capture the contextual information between segments using co-occurrence statistics from training data at different locations and scales.

2.2.1 Appearance

Our data term $\Psi_i(c_i) = p(c_i|S_i)$ depends on the texture, shape and color of the segment. To incorporate the texture and shape information, we use the Textonboost [13] code [27] with one modification. Textonboost incorporates context through the appearance of surrounding texture patches. Since we are interested in modeling context at the object level and not implicitly through features, we trained Textonboost on individual objects and not entire images, using the ground truth segmentations. Thus any contextual information captured by Textonboost from surrounding objects was removed. In our experiments 700 rounds of boosting were performed instead of 5000 as used in [13]. The resulting class likelihoods for each pixel found by Textonboost are averaged across each segment to obtain

a vector with length C equal to the number of possible classes.

To incorporate color, we train a Gaussian Mixture Model (GMM) for each class. We used 7 Gaussians per class in the three-dimensional RGB space. The likelihoods for each pixel are averaged across the segments to obtain a C length vector. In order to combine the results of Textonboost and the color GMM to obtain $\Psi_i(c_i)$, we use an approach similar to He *et al.* [12]. The two C length vectors are concatenated and passed through a multi-layer perceptron neural network with C outputs. We used 20 hidden layer nodes in our experiments with a sigmoid transfer function.

2.2.2 Context

The edge-interactions $\Phi_{ij}(c_i, c_j)$ capture the contextual information between segments S_i and S_j through co-occurrence counts given the segments' locations and scales. This is modeled as

$$\Phi_{ij}(c_i, c_j) = [\phi_{ij}(c_i, c_j) + \epsilon]^{\eta}. \tag{2.2}$$

In all our experiments, ϵ was fixed to be 1 and corresponds to a weak Dirichlet prior. η was 0.02, which controls the effect of context with respect to the data term. Further,

$$\phi_{ij}(c_i, c_j) = \kappa(c_i, c_j)\lambda_{ij}(c_i, c_j)\varphi_{ij}(c_i, c_j), \tag{2.3}$$

where $\kappa(c_i, c_j)$ captures the likelihood of classes c_i and c_j co-occurring in the image, $\lambda_{ij}(c_i, c_j)$ represents the likelihood of segments S_i and S_j co-occurring at their observed locations given assignments to classes c_i and c_j, and similarly $\varphi_{ij}(c_i, c_j)$ represents the likelihood of segments S_i and S_j co-occurring with their

observed scales given assignments to classes c_i and c_j. We describe these next.

Co-occurrence: $\kappa(c_i, c_j)$ is the empirical probability of classes c_i and c_j co-occurring in an image. This is learnt through MLE counts from the labeled training data.

Location: We model the location of a segment in an image using a Gaussian Mixture Model with $L = 9$ components. For our experiments the Gaussian means are centered in a 3×3 grid with standard deviations in each dimension equal to half the distance between the means. We define the value $\alpha_l(l_i)$ as the average likelihood of S_i's pixels being in component $l \in L$. Since most images have a horizontal layout we also tried using only 3 bins spaced vertically apart, but the results were significantly worse. The value of $\lambda_{ij}(c_i, c_j)$ is computed as

$$\lambda_{ij}(c_i, c_j) = \sum_{l_i=1}^{L} \sum_{l_j=1}^{L} \alpha_l(l_i)\alpha_l(l_j)\theta_l(l_i, l_j|c_i, c_j), \tag{2.4}$$

where $\theta_l(l_i, l_j|c_i, c_j)$ are parameters estimated from training data through MLE counts. More specifically, $\theta_l(l_i, l_j|c_i, c_j)$ is the empirical probability of the segments S_i and S_j occurring at locations l_i and l_j given their assignments to classes c_i and c_j. It should be noted that this is a joint distribution, and thus includes both the absolute location and relative location statistics i.e. $\theta_l(l_i, l_j|c_i, c_j)$ combines the information $\theta_l(l_i|c_i)$ and $\theta_l(l_j|l_i, c_i, c_j)$. Since the absolute location is measured relative to the image, the statistic $\theta_l(l_i|c_i)$ can be viewed as contextual information relative to the entire image.

Scale: The scale is defined as the proportion of the number of pixels in the segment with respect to the number of pixels in the image. As a result, the scale for each segment has a value between 0 and 1. Similar to location, we model the scale using a GMM. The GMM has $K = 4$ components with means evenly spaced between 0 and 1. The standard deviation of the components are set to half the distance between the means. We define $\alpha_s(s_i)$ as the likelihood of a segment belonging to scale s_i. $\varphi_{ij}(c_i, c_j)$ is then computed as

$$\varphi_{ij}(c_i, c_j) = \sum_{s_i=1}^{K} \sum_{s_j=1}^{K} \alpha_s(s_i)\alpha_s(s_j)\theta_s(s_i, s_j|c_i, c_j), \tag{2.5}$$

where $\theta_s(s_i, s_j|c_i, c_j)$ are parameters estimated from training data through MLE counts. Again, $\theta_s(s_i, s_j|c_i, c_j)$ is the empirical probability of segments S_i and S_j having scales s_i and s_j given their assignments to classes c_i and c_j. As with location, the absolute and relative scale statistics are both captured here.

We use Loopy Belief Propagation to perform inference on the CRF using a publicly available implementation [28]. After convergence, the label with maximum belief is assigned to the segment.

Using equation (2.3) we maintain the simplicity of the model proposed in [7], which uses just co-occurrence counts, while capturing richer information through relative location and scale statistics. The proposed model also allows for the straightforward incorporation of additional contextual information, such as relative 3D orientations if available, using the same formulation. We do not do any parameter learning to explicitly increase the likelihood of the training data under our model. Although the current treatment suffices for our purposes, explicit parameter learning such as in [7] may further boost performances.

Figure 2.3: Low resolution images from the MSRC (top) and Corel (bottom) datasets. The larger dimension is 32 pixels. The objects are often very small, for instance there are only 4 pixels in the faces in the top left image.

2.3 Results

In our experiments we use the MSRC dataset [23] and a subset of the Corel dataset [24]. The MSRC dataset contains 591 images with pixel-wise labels coming from 23 classes. Following previous works, we remove 2 classes (horses and mountain) because of very few training instances. The Corel dataset consists of 100 images with labels coming from 7 classes. As stated earlier, we work with images at their original resolution ($\sim 320 \times 320$) pixels, as well as at low resolution ($\sim 32 \times 32$ pixels). In both datasets, a random subset of 45% of the images were used for training, 10% for validation and the rest for testing, while maintaining consistent class distributions in these three sets, similar to [13]. We show sample low resolution test images from both datasets in Figure 2.3. We first describe our human studies, followed by our machine vision experiments and finally some analysis of the results obtained.

2.3.1 Human Studies

Our human studies were performed on the MSRC dataset using 11 subjects. The task assigned to them was to identify the outlined segment in the displayed image. Each subject had to complete two sessions. The first session was on the low resolution images and the second on the original images. In each session, there were three scenarios under which the subjects had to recognize the segments. The first studied appearance-based recognition by only displaying the segment to be recognized without the rest of the image, Figure 2.1(a, d). The second studied blind recognition in which the subject was shown the image with the pixels removed from the segment to be recognized, Figure 2.1(b, e). The final scenario displayed the entire image allowing the subject to use both appearance and contextual information for recognition, Figure 2.1(c, f). In each scenario the images were displayed with the segment outlined, as well as without the segment outlined to avoid distraction. For low resolution images, the images were displayed at four different scales (32×32, 64×64, 128×128 and 256×256) using bicubic interpolation so that the subjects could focus on whichever scale they desired, without increasing the amount of information being displayed [21]. The list of possible classes from which the subjects could choose was displayed below the images, as shown in Figure 2.4. Each subject was asked to recognize 70 segments for each scenario for each resolution (a total of 420 segments per subject). The segments to be recognized were selected randomly from a total of 650 segments in 265 images (per resolution) from the MSRC dataset. On average, subjects took 35 minutes to complete the entire study. The segment boundaries were marked using the ground truth segmentations provided with the

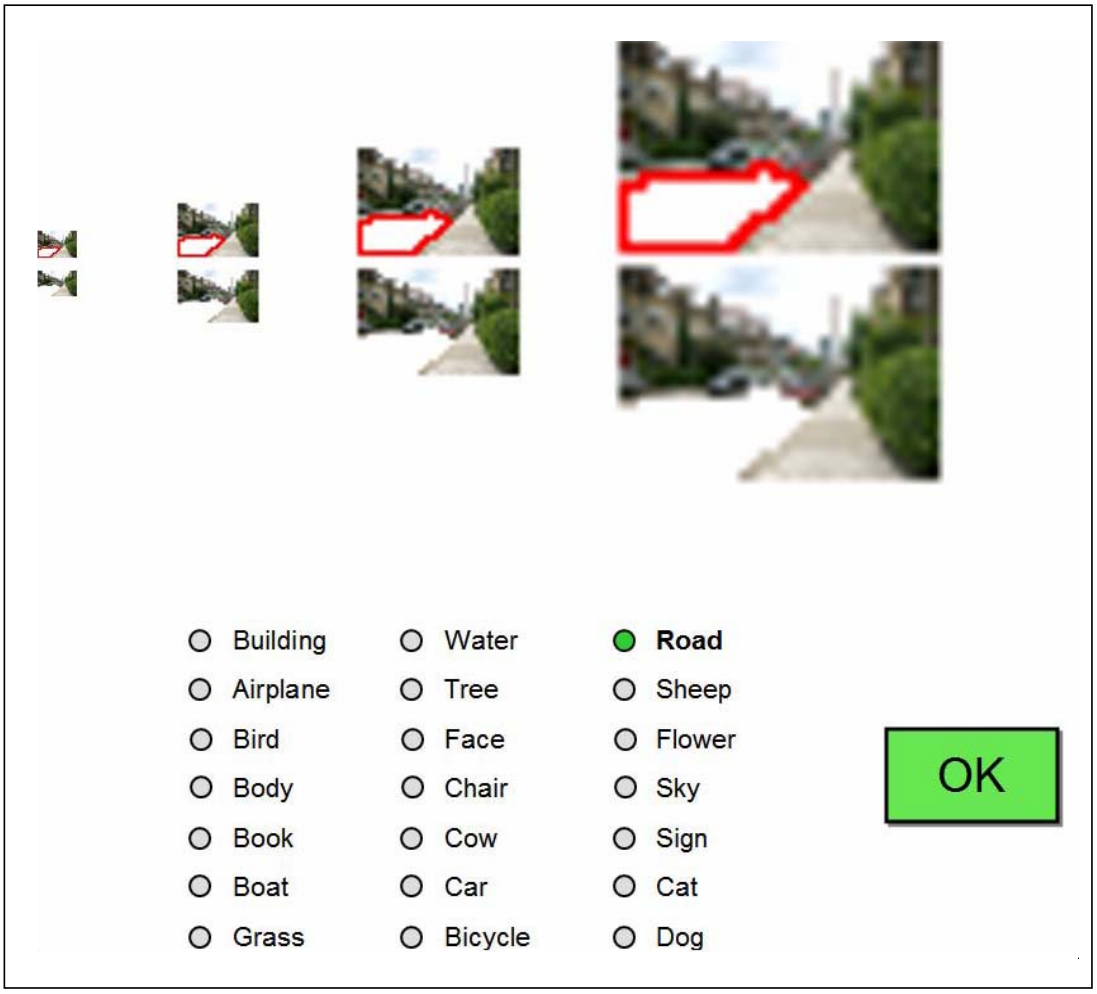


Figure 2.4: A snapshot of the interface used for human studies on low resolution images for blind recognition.

MSRC dataset.

Human accuracies have been studied in low resolution images for face recognition [29, 30], scene recognition [31, 32, 21] and more recently for object detection and segmentations [21]. However, separating the roles of context from that of appearance as the amount of appearance information varies has not been studied.

The accuracies of the subjects, computed as average class-wise accuracies, are shown in Figure 2.5 and Table 2.1. There are several observations we can make.

First, the need for context is minimal in the original high resolution images. Appearance alone performs at 96% accuracy with context increasing performance by 2%, which is below statistical significance. Secondly, appearance provides less information in low resolution images as seen by the drop in accuracy from 96% to 66%. Interestingly, blind recognition using context alone provides a similar accuracy of 67% for low resolution images. The combination of appearance and context increases accuracy by a statistically significant amount to 89%. This is in agreement with Torralba *et al.*'s observations that human recognition in 32×32 images does not reduce drastically as compared to full resolution images, and we demonstrate here that this is due to inclusion of context. These experiments further support the notion that low resolution images are an interesting venue for modeling context, where the need for context is important.

It should be noted that the subjects were given a choice of 21 possible category labels. Experiments in which the set of labels is unknown and determined by the subject may yield different results. For some objects the segments are not exact so small amounts of surrounding information, such as grass, may be present for the appearance only tests. Finally, for the task of blind recognition the information inside the segment was removed. However, the rough shape of the segment was still visible and in some cases can supply appearance based information. As a result, the accuracies of the blind recognition tests may be artificially high.

2.3.2 Machine Experiments

We replicate the human studies in our machine experiments. For consistency with the human studies, recognition was performed on the ground truth segmentations

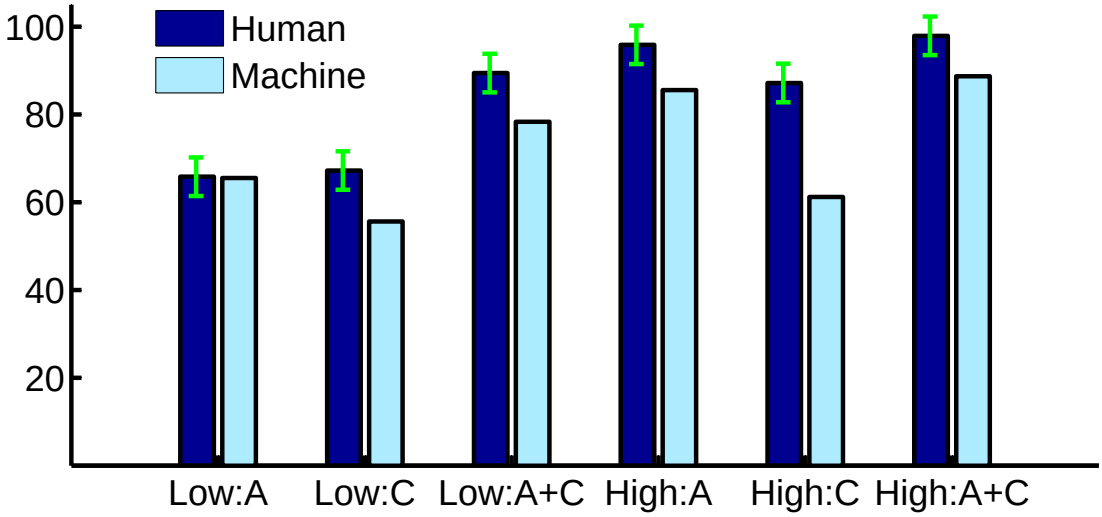


Figure 2.5: The recognition accuracies of human subjects and machine on low and high resolution images using appearance alone (A), blind recognition using context alone (C) and both appearance and context (A+C). The error bars are also indicated for human accuracies.

(later results use automatic segmentation). In the appearance-only scenario, the MAP estimates of the data terms were used to label the segments. For blind recognition, the data term corresponding to the segment to be recognized was set to a uniform distribution before running inference on the CRF.

The results obtained on the MSRC dataset are shown in Figure 2.5 and in Table 2.1 with results on the Corel dataset. For consistency, we use the same 265 images of the MSRC dataset for testing as were used in the above human studies. The results on other random splits are consistent with those shown here. We see very similar trends in the machine numbers as with those from the human studies. With low resolution images, we see that combining appearance and context significantly boosts performance over each individually, to 78% for MSRC and 87% for Corel. Tests on images with their original resolution show a comparatively smaller, however non-trivial boost in performance. It is interesting

Table 2.1: Machine and human accuracies on MSRC and Corel datasets

	A	C	A+CO	A+CO+L	A + C
MSRC					
Low	65.51	55.62	71.91	76.65	78.33
High	85.55	61.21	87.04	87.73	88.65
MSRC Human					
Low	65.81	67.23	-	-	89.42
High	95.85	87.12	-	-	97.90
Corel					
Low	74.57	62.77	86.19	86.64	87.29
High	91.23	70.84	97.38	98.23	98.16

A → appearance; C → context → co-occurrence CO + relative location L + relative scale

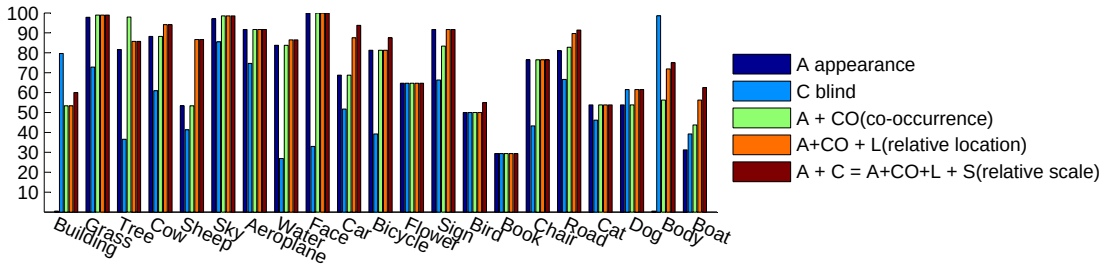


Figure 2.6: Average accuracies for the 21 categories in the MSRC dataset using appearance alone, using blind recognition with context alone, and using subsequently more complex context models with appearance.

to note that identical context models were used for images of both resolutions, while the appearance information was trained separately.

Different sources of context: We present some analysis to evaluate the contribution of the different forms of context (co-occurrence CO, relative location L and relative scale S). Figure 2.6 shows the per class accuracies on low resolution images using only appearance, and subsequently adding the three forms of context. We can see that different object categories benefit from different forms

of context. Some categories such as books and chairs do not receive any benefit from context due to peculiarities of the dataset, such as they rarely co-occur with other objects, Figure 2.7. Categories such as body and boat gain significantly from context. Their appearance cues are very weak (0% in the case of body), but they are very strongly associated with other categories (Face and Water respectively) whose appearance cues are quite reliable. In fact, for some categories such as Body and Building, blind recognition performs much better than appearance information alone as well as combined appearance and context. In several categories, relative scale does not provide a boost in performance. This may be due to lack of scale related dependencies due to inherent semantics of the categories, or due to depth variations of the objects across images, to which our scale measure is not invariant. This lack of dependency is automatically learnt by our model. In some categories, albeit rarely, certain forms of context hurt performance. This may be attributed to a category's strong dependence on categories with poor appearance cues. For instance, Sign commonly co-occurs with Building whose appearance term has 0% accuracy.

Figure 2.7: Images in the MSRC dataset containing books. They occur at similar locations across images, and rarely interact with other categories. Contextual information does not boost the performance of such categories.

Average class-wise accuracies using both low and high resolution images from the MSRC and Corel datasets for each of the different forms of context are summarized in Table 2.1. The Corel dataset has fewer classes and the only prominent

interactions are the co-occurrence of polar bears with snow, and rhinos/hippos with water. Hence, while co-occurrence gives a significant boost in performance on the Corel dataset, relative location and relative scale do not. For MSRC, which is a richer dataset, all forms of context give a significant boost on low resolution images.

In Figure 2.8 several examples are shown where different types of context helped recognition. Let us consider the last example, where the test image contains Tree, Car, Road and Sky. The appearance alone labels the objects as Tree, Cat, Road and Sky, but the very low likelihood of finding a Cat on the Road along with Tree and Sky made the co-occurrence information flip the label of the Cat to a Building. The location of the Building seems consistent with respect to the Tree, Road and Sky - so the relative location information left the labels untouched. However, the relative scale information discarded the possibility of the Building being so small with respect to the Sky, Tree and Road, and flipped the label of the Building to Car - which matches the ground truth labeling. Other intuitive examples are shown in Figure 2.8 as well. Examples of incorrect labels provided by the context model are shown in Figure 2.9.

Comparison with other works: We also perform the same experiments with automatic segmentations. We use the Felzenshwalb and Huttenlocher [33] segmentation algorithm (example segmentations in Figure 2.10). Our results are shown in Table 5.1 along with a comparison to results from previous works when available. In addition to the segment-wise accuracies metric we have used so far, we report pixel-wise accuracies as well. To obtain a pixel-wise label map from our model, all pixels falling within a segment were assigned the segment's predicted la-

Table 2.2: Comparisons of accuracies

	MSRC		Corel	
	pixel	segment**	pixel	segment
Shotton *et al.* [13]	58(72)	– (71)	– (75)	–
Yang *et al.* [34]	62(75)	–	–	–
Verbeek *et al.* [35]	64(74)	–	–	–
He *et al.* [12]	–	–	81(80)	–
He *et al.* [36]	–	–	– (81)	–
Rabinovich *et al.* [7]	–	– (68)	–	–
High	85(91)	84(89)	94(93)	95(93)
Low	81(83)	77(81)	86(86)	85(84)

Different splits may have been used for training and testing data ** Segment-wise accuracies may not be directly comparable because the exact settings under which the accuracies were computed may differ

bel. For our own algorithm, we report results on original (high) resolution images that all other works use, as well as on low resolution images. We report average class-wise accuracies, as well as overall accuracies (within parentheses). Even when using low resolution images, our algorithm outperforms previous works on these datasets.

We believe this is due to several reasons. He *et al.* [12] and Shotton *et al.* [13] make decisions at the level of pixels or small patches, while we do so on segments which requires only a few decisions per image. This also allows us to train on segments making the training information more reliable due to inherent aggregation and grouping. Our explicit use of color was found to give a significant boost in performance. A notable observation is that the difference between our average class-wise accuracies and overall accuracy is not very large.

2.4 Discussion

In this section we draw attention to some interesting points of discussion.

2.4.1 Organizing categories according to context

The different categories in a dataset can be organized according to their pairwise contextual relationships, to observe groupings among related categories. We use the negative logarithm of the normalized co-occurrence matrix among the 21 categories of the MSRC dataset as a dissimilarity measure among these categories. We use Multi Dimensional Scaling to project these 21 categories on a 2D plane. The resultant visualization is shown in Figure 2.11. We also use normalized cuts on the dissimilarity matrix to cluster these categories. In Figure 2.11, an edge is drawn between two categories if they were assigned to the same cluster, and had a dissimilarity measure lower than the average dissimilarity across all categories.

We see that semantically meaningful categories are placed closed to each other such as Water-Boat and Face-Body. The clusters also corresponding to groups of categories that tend to co-occur in the real-world. Interestingly, Cow, Sheep, Airplane and Grass were all assigned to the same cluster, however the edges indicate that the strong interactions hold among the objects and Grass individually, and now amongst each other. This visualization also allows us to identify peculiarities of the dataset. For instance, the categories Cat or Book or Sign not being clustered with any category indicate weak contextual ties. Our analysis indicated that the projection into 2D has several misleading distances. For instance, Flower and Dog are shown at the same distance as Sheep and Grass, however the co-occurrence counts of these pairs are significantly different, with Sheep and

Grass co-occuring very frequently. These relationships are better demonstrated through the normalized cuts clustering.

2.4.2 Accuracies on a dataset by *chance*

To analyze the amount of contextual information in a dataset, an interesting metric is to look at what recognition accuracy corresponds to *chance* as the different forms of context are incorporated. For instance, if we had no information, in a 21 class problem, chance would be 1/21 i.e. about 5%. However, if we analyze the location statistics of the different categories, and given a segment to be recognized, use that location information to make our best guess, and our *chance* accuracy is now higher. We still refer to this as chance because no appearance information or other intelligent machinery has been used, we are simply making our best *guess* blindly. Similar other statistics such as scale, and location and scale combined can be extracted from training data to evaluate what recognition rates can be achieved just by chance in a given dataset, which sheds better light on how good state-of-the-art algorithms are compared to this chance.

The blue bars in Figure 2.13 indicate the recognition rates we get by chance when classifying each segment/object in the MSRC dataset using uniform prior, occurrence-based prior, location-based prior, scale-based prior, and location-and-scale-based prior. As expected, with more information, the recognition rate of chance increases, upto about 32%, much higher than the 5% we may be inclined to consider for a 21 class problem, or even 14% given the distribution of classes in this dataset.

This was the task of classifying each segment individually. This scenario may

be deceiving. For instance, when considering only occurrence prior, we would classify every segment as the the same class (the most popular one - such as grass in the case of the MSRC dataset), and get 14% recognition rate, without having really *understood* any image well. So perhaps a more interesting metric is to classify higher order (groups of) objects correctly. We consider the scenario where the task is to classify a pair of objects correctly (getting even one object incorrect, is an incorrect classification of the pair). In this case, relevant information would be co-occurrence information, relative location statistics, relative scale statistics and relative location and scale statistics combined (similar to our context model). Since the number of classes a pair of objects can be assigned to is much larger (441 for the MSRC dataset), the uniform prior has a much lower recognition rate (0.2%). The red bars in Figure 2.13 shows the recognition rates of classifying pairs of objects using these different priors. Again, with more priors incorporated, chance goes up significantly, upto about 17% even for pairs of objects.

For sake of completeness, we show the accuracy for classifying individual objects using the optimal strategy for classifying pairs, and vice versa, as the red and blue bars in Figure 2.12 and 2.13 respectively. As would be expected, these accuracies are lower than the optimal strategy for the particular task.

2.4.3 Humans vs. Machine

We analyze some commonalities and discrepancies between the behavior of humans and machines in incorporating context into recognition. The four categories from the MSRC dataset that got the highest boost in performance on low resolution images by incorporating context for the human subjects were found to be

Body, Face, Water and Boat with Body and Face, and Water and Boat being complementary categories. The top four categories for the machine were Body, Boat, Building and Sheep, but not Face and Water. This is due to the fact that the appearance based recognition for Body and Boat were low (0% and 30%) while Water and Face were very high (85% and 100%), leaving little room for further improvement.

Figure 2.14 shows the (normalized) confusion matrices on the MSRC dataset. We now analyze the similarities in these recognition performances for humans and machines

We sort all off-diagonal elements of the confusion matrices in descending order (in decreasing order of confusion observed between these two categories), for humans and machines, for all four scenarios (using appearance information in low resolution images, appearance and contextual information in low resolution images, and their counterparts in high resolution). We consider the top n confusing category pairs for the human studies, and compare those with the top n confusing category pairs for the machine, and determine the number of common category pairs (the largest number of common category pairs can be n). We vary n, and the obtained plot is shown in Figure 2.15. The red curve shows the upper bound (identity), while the green curve indicates the curve obtained if the machine ranks were random (no correlation with human ranks). We find a strong correlation between the human and machine rankings in our experiments (blue curve).

Similarly we compute a ranking for category pairs according to the decrease in the corresponding confusion by incorporating contextual information to the low resolution appearance information. As shown in Figure 2.16, we find that the machine rankings are correlated with the human rankings to some extent. We also

show the correlation in rankings of category pairs according to the decrease in the corresponding confusion by incorporating high resolution appearance information.

We also compare the rankings of category pairs within the human studies (and machine experiments), to see if the categories that benefited most from incorporating context also benefited from incorporating high resolution information. The resultant curves are seen in Figure 2.17. We find that the two are in fact correlated, which seems to indicate that the category pairs with low accuracies using low resolution appearance information, can benefit from additional information - be it in the form of context, or high resolution appearance information. And as our earlier experiments show, once we incorporate high resolution information, context does not provide further boosts in performance. This once again stresses the potential of using low resolution images to model context, as opposed to high resolution images.

2.4.4 Improving features or context models?

We explore the question "Do we need to improve our data terms further or our context models to achieve close to human accuracies?" Looking at the MSRC high resolution results in Figure 2.5 we find that machines are lagging significantly behind on using appearance information alone. For low resolution images, in which the appearance only tests between humans and machines are similar, the use of context helps humans significantly more. Thus it appears improvements on using both appearance and contextual information need to be made to match the performance of humans. Since tests using only appearance information are similar for humans and machines on low resolution images, this task provides a

good scenario for evaluating context models.

2.4.5 Context as representing the structure in the world

As we see in our results, the gain from context is certainly a characteristic of the dataset. The more complex a scene, the greater the likelihood of it benefitting from context. As the complexity and number of objects increases, obtaining training datasets with sufficient information will be more difficult. Means of learning context from outside sources such as Google Sets as recently proposed by Rabinovich *et al.* [7] or extensive collection of image data such as LabelMe [37] may need to be explored. The easy availability of training data is needed to learn the generic structure of our world, as opposed to potential peculiarities of a dataset.

2.5 Conclusion

In conclusion this chapter contains two main contributions. First, we propose a model for context that includes relative location and scale information, as well as co-occurrence information. Our results show relative location and scale contextual information produces state-of-the-art performance on both the MSRC and Corel datasets even with low resolution images. Second, we explore the tradeoffs of appearance and contextual information using both low and high resolution images in human and machine studies, and find that high resolution images do not benefit much from the incorporation of contextual information. Low resolution images provide an appropriate venue for exploring the role of context since recognition based on appearance information alone is limited.

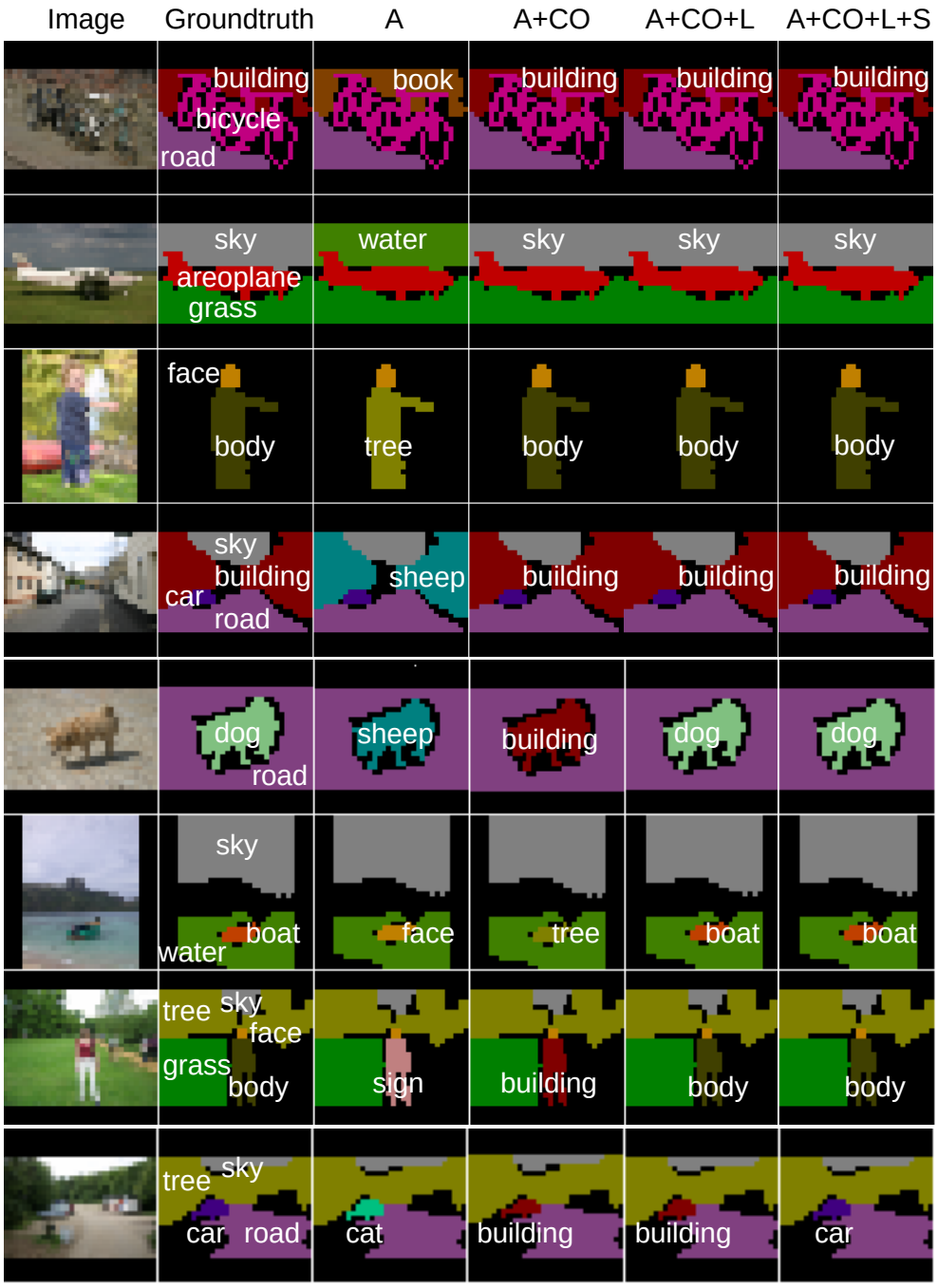


Figure 2.8: Illustrations of the effects of different forms of context on recognition. A → appearance, CO → co-occurrence, L → relative location, S → relative scale. (Viewed better in color)

Figure 2.9: Illustrations of incorrect labelings provided by the context model. (Viewed better in color)

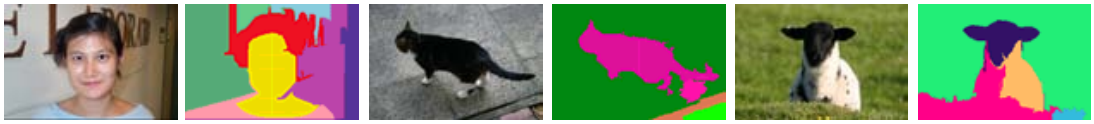


Figure 2.10: Illustrations of automatic segmentaitons

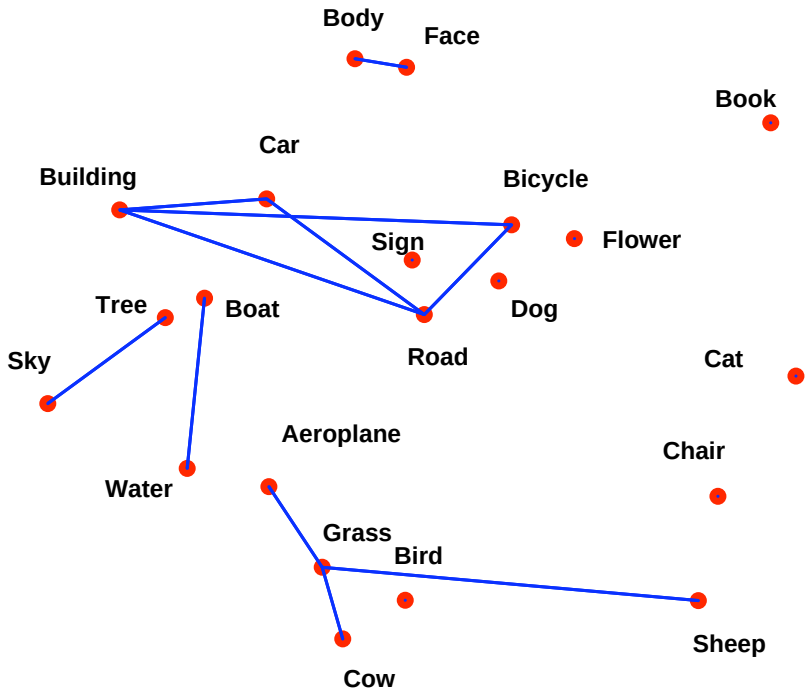


Figure 2.11: The 21 categories of the MSRC dataset projected on a 2D plane where smaller distances between points (approximately) reflect high co-occurrence in images. The categories connected with an edge were assigned to the same cluster by normalized cuts, and had high co-occurrence.

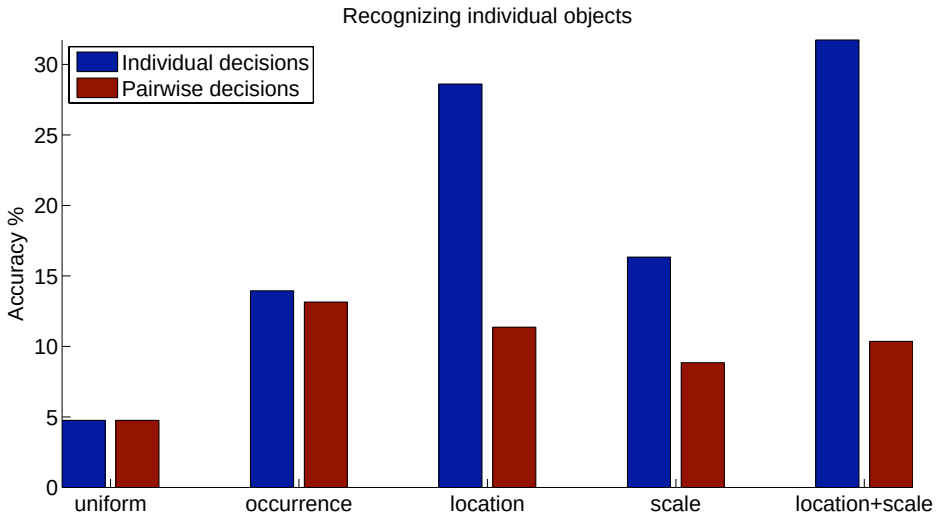


Figure 2.12: Different baselines for *chance* in the MSRC dataset for recognizing individual objects/segments in an image

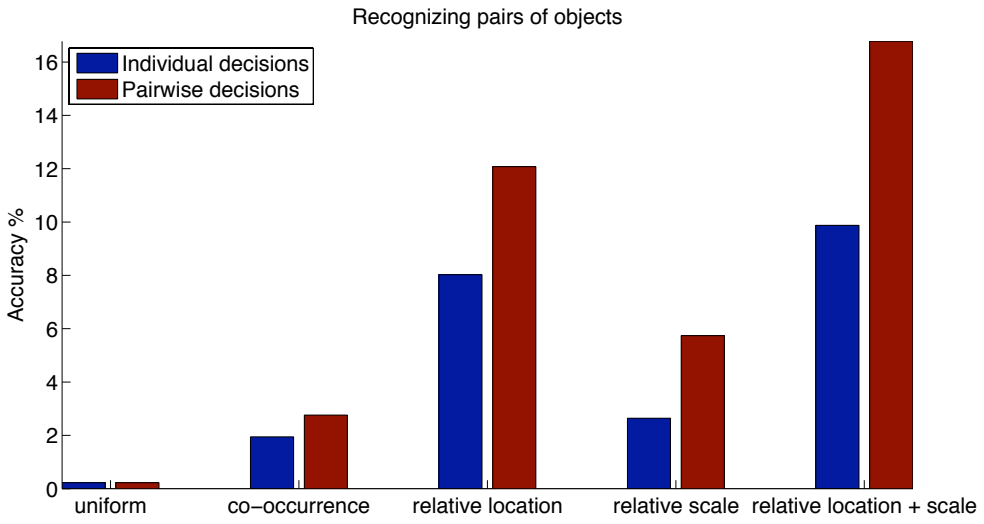


Figure 2.13: Different baselines for *chance* in the MSRC dataset for recognizing pairs of objects/segments in an image

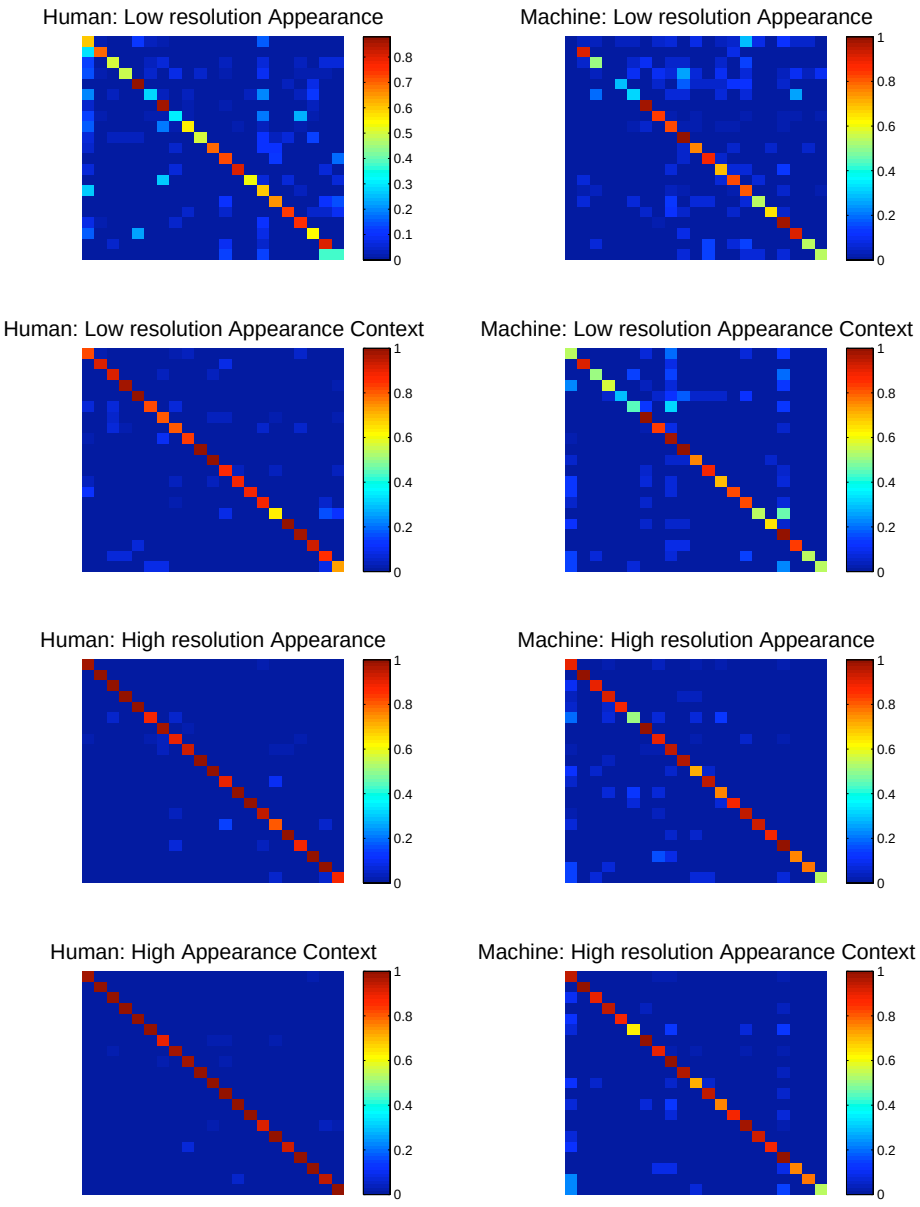


Figure 2.14: Confusion matrices of the human studies and machine experiments on the MSRC dataset using the ground truth segmentations

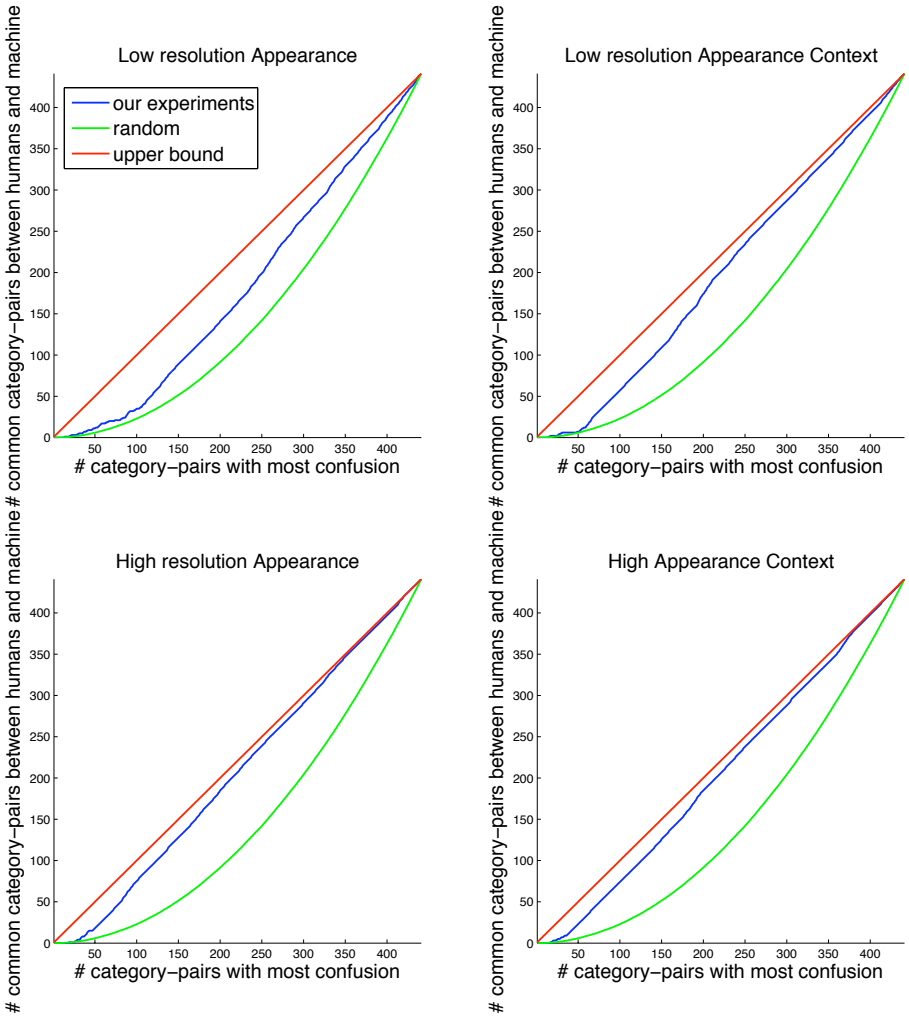


Figure 2.15: Comparing the human and machine rankings of category pairs to indicate the confusion between these categories

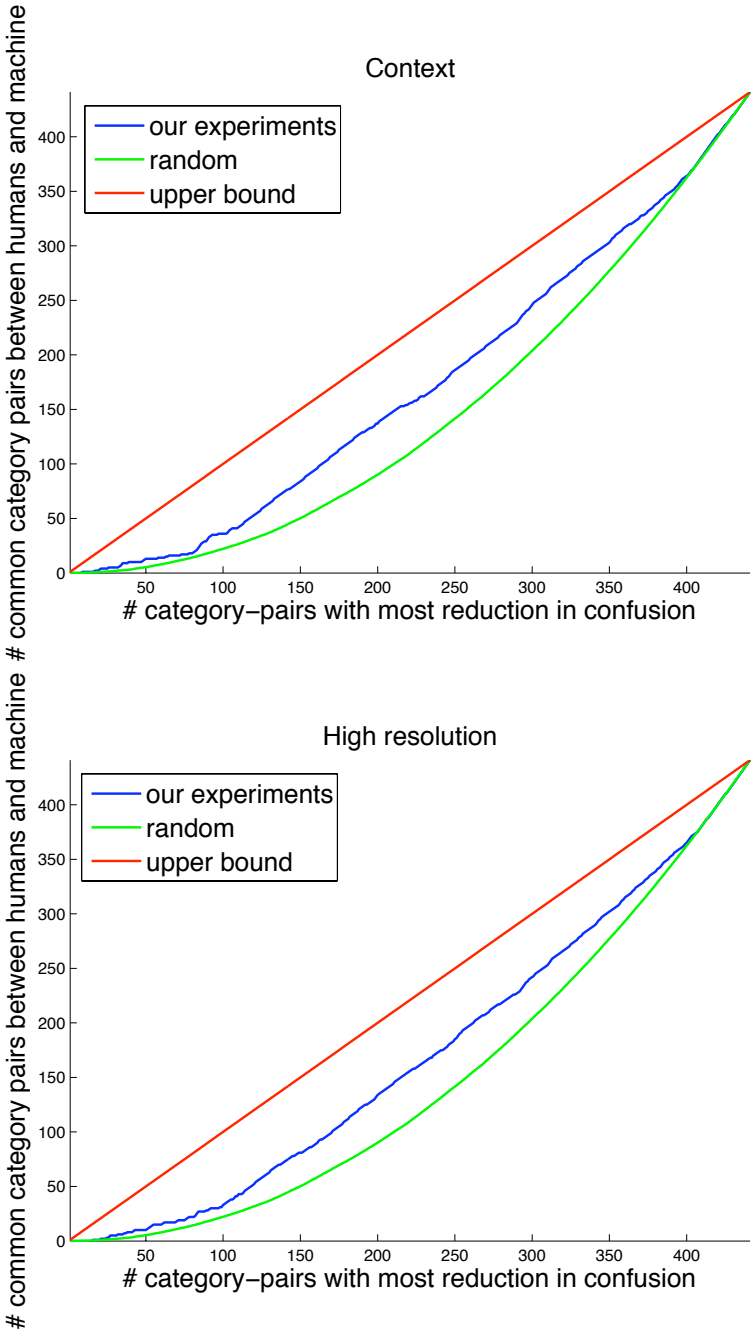


Figure 2.16: Comparing the human and machine rankings of category pairs to indicate the benefit (reduction in confusion) by incorporating context and high resolution appearance information

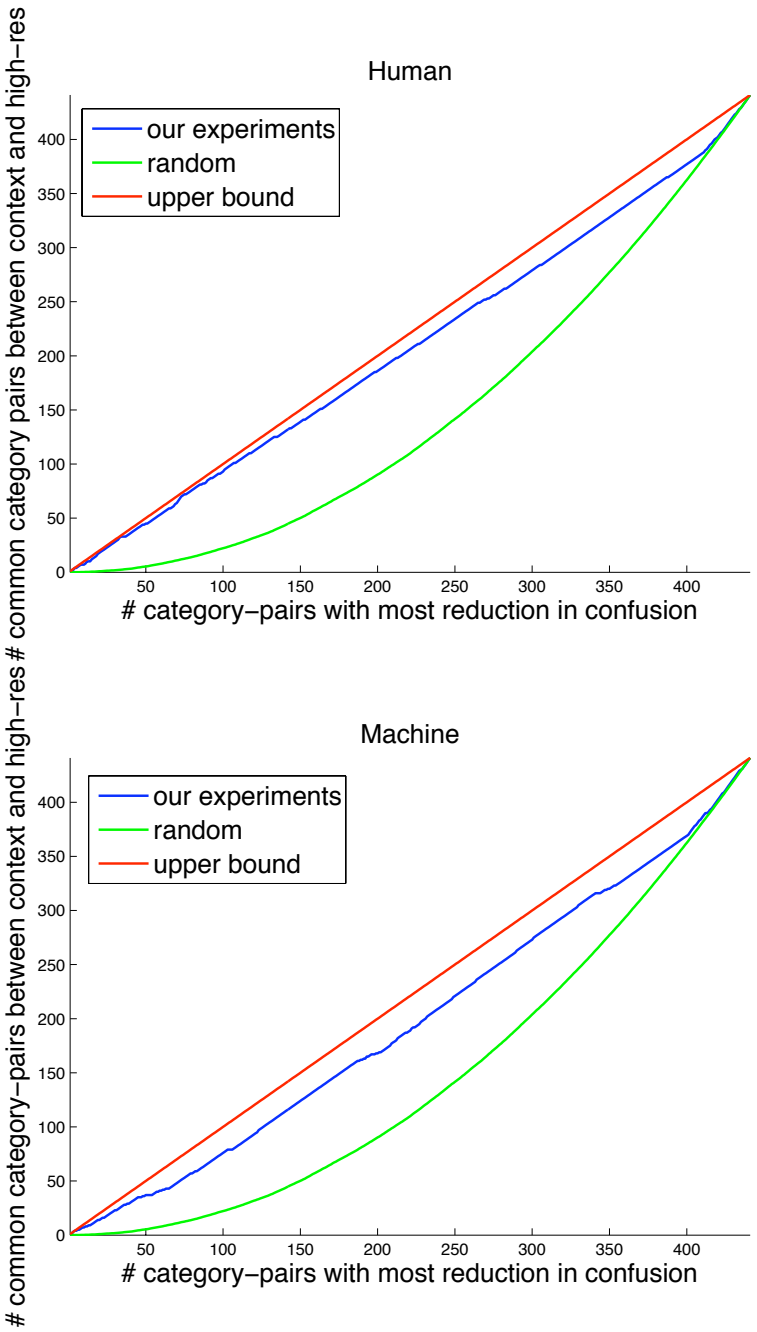


Figure 2.17: Comparing the benefits of incorporating context to those of incorporating high resolution information, within human studies and the machine experiments.

Chapter 3

For What: Determining Low-Level Patch Saliency

Summary

The increased use of context for high level reasoning has been popular in recent works to increase recognition accuracy. In this chapter, we consider an orthogonal application of context. We explore the use of context to determine which low-level appearance cues in an image are salient or representative of an image's contents. Existing classes of low-level saliency measures for image patches include those based on interest points, as well as supervised discriminative measures. We propose a new class of unsupervised contextual saliency measures based on co-occurrence and spatial information between image patches. For recognition, image patches are sampled using a weighted random sampling based on saliency, or using a sequential approach based on maximizing the likelihoods of the image patches. We compare the different classes of saliency measures, along with a

baseline uniform measure, for the task of scene and object recognition using the bag-of-features paradigm. In our results, the contextual saliency measures achieve improved accuracies over the previous methods. Moreover, our highest accuracy is achieved using a sparse sampling of the image, unlike previous approaches whose performance increases with the sampling density.

3.1 Introduction

Determining image patches of high saliency has recently received significant attention. The goal of saliency detection is to identify the image patches that are most informative of the image contents. A standard method for finding these patches is the use of interest point detectors based on local low-level image statistics [38, 39, 40, 41, 42, 43]. Another class of saliency measures are discriminative in nature [44, 45, 46, 47, 48, 49], where a patch is considered salient if it is informative from a classification perspective. The usefulness of a patch may be based on the mutual information between the presence of the patch and the scene categories [44] or the probability of misclassification of a patch [45]. Using these techniques, a relatively small number of patches can be sampled while still achieving high recognition accuracy.

In this chapter, we explore the use of contextual information that is typically used for higher level reasoning for the low-level task of selecting informative or salient patches in an image. We consider a patch to be salient if it is predictive or representative of the other patches in the image. The relationships of image patches are modeled using co-occurrence and spatial information. Unlike previous saliency measures that rely only on local information, our approach incorporates

contextual information using the patch statistics across the entire image. For recognition, the sampling of patches is performed using weighted random sampling based on patch saliency. In addition, we propose a sequential sampling approach based on increasing the maximum likelihood of patches given the set of previously selected patches.

Our saliency measure is evaluated within a bag-of-features framework [50, 51, 52, 53, 54]. This simple approach has shown good performance in a variety of recognition tasks and allows us to focus on the specific contributions of our chapter. The bag-of-features approach consists of three components: a method for sampling patches from an image, a method for assigning the patches to a discrete patch vocabulary, and a method for classifying the resulting global descriptor. In this chapter, we only address the first task of patch sampling, and use standard approaches for the other two components. A vocabulary of patch appearances, to which each patch is assigned, is constructed using K-means clustering [55, 56]. Classification is accomplished using an SVM classifier over the histogram of patch assignments [44, 52].

We compare our proposed contextual saliency measures to a variety of existing measures including interest points, discriminative approaches and random sampling. These measures are evaluated on both scene and object recognition tasks. In contrast to previous results that show recognition accuracy increases with the density of the sampling [44], the contextual measures achieve maximal accuracy using a sparse sampling. Moreover, in our experiments the accuracy of using contextual measures with sparse sampling is better than dense sampling using other methods.

The rest of the chapter is organized as follows: Previous works are discussed in

the following section. We describe our proposed contextual saliency measures in Section 3.3 and sampling methods in Section 3.4. In Section 3.5 we briefly describe the existing saliency measures used for comparison, followed by a description of the experimental setup in Section 3.6. Results and some discussion are provided in Section 3.7 and conclusions in Section 3.8.

3.2 Previous Work

Several works have explored the role of saliency measures for classification tasks. Nowak *et al.* [44] compare the interest operators to random dense sampling and find that random sampling performs comparable or superior to interest operators. Jurie *et al.* [57], apart from proposing a novel clustering approach to form codebooks, evaluate a discriminative saliency measure used for the feature selection problem. They find that when using smaller codebooks discriminative feature selection can be used to improve accuracies. However, using the full codebook for classification typically resulted in better performance. A related class of works [45, 58, 59] is visual search, where similar notions of saliency are important. The essence of visual search lies in the notion of active exploration, in which saliency maps are dynamically updated or areas are marked for further exploration.

Most existing approaches [38, 41, 52, 54, 55, 60, 61, 56] for selecting a sparse set of image patches are based on interest point detectors [38, 39, 40, 41, 42, 43]. These include those based on edge cornerness [39], difference of Gaussian convolutions [38], stable extremal regions [42] or local entropy [40]. While these measures are useful for obtaining reliable correspondences or matching, they do not relate

directly to image understanding via classification or recognition. The strategies of dense sampling [50, 53, 62] or random sampling [49] have shown to provide comparable or even better performance than interest points [44]. Biologically inspired saliency measures following the "feature integration theory" [63] extract regions of the image that stand out from their surrounding as being salient [64, 65]. [40, 66] are based on a similar notion and [67] consider features to be salient if they are rare. While this is a plausible explanation to predict task-independent attention, they do not take into account task-dependencies. Walther *et al.* [68] incorporate such task dependencies and combine the biologically plausible saliency map of [65] with interest point operators [38] to show improved performances.

Using high-level contextual information for better image understanding has received significant attention in recent works [8, 9, 10, 11, 18, 19, 7, 69, 17]. Most of these approaches use context as a post-processing step to prune out false positives [8, 9], aid in detection by eliminating unlikely locations of objects [8, 10, 11, 18, 19], or ensure semantically consistent labels to regions of an image [8, 9, 7, 69, 17].

3.3 Proposed Contextual Saliency Measures

Our goal is to select a sparse set of image patches that are most informative for classification. We propose that the patches, which are representative or predictive of other patches in the image, are also the patches most useful for classification. We measure the predictiveness of a patch using a contextual saliency measure based on co-occurrence and spatial information. As stated earlier, we examine our measure of saliency within the bag-of-features framework. In this framework,

classification is achieved by selecting a set of image patches and assigning them to codewords. A histogram of codewords is then constructed and used for classification. In this chapter, we address the first task of selecting image patches. We describe the standard method of K-means for codebook creation and Support Vector Machines for classification in Section 3.7.

For each image patch x_i, we compute a patch descriptor y_i. This descriptor can vary based on the application and properties of particular datasets. In this chapter we examine two descriptors. The first is a 4×4 vector of average color values over a patch. This descriptor is useful for scenarios in which color information is important, such as in scene recognition. For object recognition in which edge information is more useful than color, we use the standard SIFT descriptor [38].

The codebook W consists of m descriptor templates. Each patch x_i in an image is assigned to a codeword w_a in the codebook. These assignments may be soft or hard with α_{ia} being the probability of patch x_i being assigned to codeword w_a:

$$\alpha_{ia} = p(y_i|w_a) = \frac{1}{Z}\mathcal{N}(y_i; w_a, \sigma_w)$$

$\mathcal{N}$ is the standard normal distribution with mean w_a and variance σ_w. The value of Z is set so that the values of α_{ia} sum to one for all a, i.e. $\sum_{a=1}^{m} \alpha_{ia} = 1$. For hard assignments $\alpha_{ia} = 1$ for the codeword w_a that lies closest to y_i and $\alpha_{ia} = 0$ otherwise.

For each patch x_i in an image, we want to assign a saliency measure $\mathcal{S}_i$. In the following two sections we propose two saliency measures based on contextual information.

3.3.1 Occurrence-based Contextual Saliency

Our measures of contextual saliency are based on how well each individual patch can predict the occurrence of other patches in the image. Our first saliency measure $\mathcal{S}^o$ uses co-occurrence information between codewords in images. Given a set of n patches in an image, we define the saliency of a patch x_i equal to the average likelihoods of the image patches conditioned upon y_i.

$$\mathcal{S}_i^o = \frac{1}{n} \sum_{j=1}^{n} \sum_{a=1}^{m} \alpha_{ia} p(x_j|w_a) \tag{3.1}$$

The value of $p(x_j|w_a)$ is computed by marginalizing over all possible codeword assignments for x_j

$$p(x_j|w_a) = \sum_{b=1}^{m} \alpha_{jb} p(w_b|w_a) \tag{3.2}$$

The value of $p(w_b|w_a)$ is the empirical conditional probability of observing codeword w_b given the codeword w_a has been observed somewhere in the image. These are learnt through MLE counts from the training images. Given hard assignments of patches to codewords, the two summations over m can be removed from equations (3.1) and (3.2).

Computing the above measure can be computational expensive, especially if the codebook size and number of patches is large. One method for reducing the computational complexity is to rearrange equations (3.1) and (3.2) as:

$$\mathcal{S}_i^o = \sum_{a=1}^{m} \alpha_{ia} \frac{1}{n} \sum_{j=1}^{n} \sum_{b=1}^{m} \alpha_{jb} p(w_b|w_a) \tag{3.3}$$

The value $\Phi_a = \frac{1}{n} \sum_{j=1}^{n} \sum_{b=1}^{m} \alpha_{jb} p(w_b|w_a)$ can then be pre-computed for each a,

resulting in:

$$\mathcal{S}_i^o = \sum_{a=1}^{m} \alpha_{ia} \Phi_a \tag{3.4}$$

3.3.2 Location-based Contextual Saliency

The previous contextual saliency measure was based solely on co-occurrence information without knowledge of the patch's location. In this section we propose a saliency measure that includes spatial information. The location of a patch in an image is modeled using a Gaussian Mixture Model with $c = 9$ components. For our experiments the Gaussian means are centered in a 3×3 grid evenly spaced across an image with standard deviations in each dimension equal to half the distance between the means. We define the value β_{iu} as the likelihood of x_i belonging to component l_u of the GMM, $u \in \{1, \ldots, c\}$, and $\sum_{u=1}^{c} \beta_{iu} = 1, \forall i$.

Similar to equation (3.1), we define our location-based contextual saliency measure $\mathcal{S}^l$ as

$$\mathcal{S}_i^l = \frac{1}{n} \sum_{j=1}^{n} \sum_{a=1}^{m} \sum_{u=1}^{c} \alpha_{ia} \beta_{iu} p(x_j | w_a, l_u) \tag{3.5}$$

The value of $p(x_j | w_a, b_u)$ is computed as

$$p(x_j | w_a, b_u) = \sum_{b=1}^{m} \sum_{v=1}^{c} \alpha_{jb} \beta_{jv} p(w_b, l_v | w_a, l_u) \tag{3.6}$$

The value of $p(w_b, l_v | w_a, l_u)$ is the empirical conditional probability of observing word w_b at location l_v given word w_a occurred at location l_u. These are learnt through MLE counts from the training images.

Similar to equation (3.4), we may pre-compute the values

$$\Psi_{au} = \frac{1}{n}\sum_{j=1}^{n}\sum_{b=1}^{m}\sum_{v=1}^{c}\alpha_{jb}\beta_{jv}p(w_b, l_v|w_a, l_u)$$

and find $\mathcal{S}_i^l$ as

$$\mathcal{S}_i^l = \sum_{a=1}^{m}\sum_{u=1}^{c}\alpha_{ia}\beta_{iu}\Psi_{au} \tag{3.7}$$

Since our proposed saliency measures are dependent on the codeword assignments of other image patches, a significant number of patches need to be sampled from the image for the measures to be reliable. However, we will only select a subset of these image patches for use in classification. While it may seem advantageous to use all the patches for classification, as we show later in our results, using a subset of the patches can actually lead to improved recognition rates.

As can be seen, there is no dependence of the saliency measures $\mathcal{S}^o$ or $\mathcal{S}^l$ on the class labels of the images, making the proposed contextual saliency measures unsupervised.

3.4 Sampling Strategies

Using the equations above we can compute a saliency measure for each patch in an image. In this section we discuss three methods for selecting a subset of these patches for use in classification. Let us assume s patches are desired for classification out of a possible n.

3.4.1 Sampling by Sorting

A naive approach to sampling is to pick the s patches with highest saliency. However, due to strong correlations in natural images, neighboring patches often have similar appearances, and would hence share similar saliency values. The result of using this technique is many neighboring patches being selected that convey similar information. As a consequence, classification rates may suffer.

3.4.2 Random Sampling

One method to reduce the odds of sampling neighboring or redundant patches is to use a weighted random sampling. The saliency map may be normalized to form a distribution over the patches from which samples can be drawn. This allows for patches with higher saliency to be sampled with a higher probability, without any one region dominating. This allows for a good balance between exploiting the highly salient regions, and exploring the rest of the image for other salient regions.

3.4.3 Sequential Sampling

The last strategy sequentially selects patches by considering the patches previously selected. Specifically, we pick the patch that is most predictive of the patches that were not highly likely given at least one of previously picked patches. Let us consider the saliency measures of equations (3.1) and (3.5), which compute the probability of $p(x_j|x_i)$ equal to $\sum_a \alpha_{ia} p(x_j|w_a)$ and $\sum_a \sum_u \alpha_{ia}\beta_{iu} p(x_j|w_a, l_u)$ respectively. Then given a set of previously picked patches $\{\acute{x}_1, \ldots, \acute{x}_t\}$ we com-

pute our saliency measure as

$$\mathcal{S}_i(\acute{x}_1, \ldots, \acute{x}_t) = \frac{1}{n} \sum_{j=1}^{n} \max\left(p(x_j|x_i), p(x_j|\acute{x}_1), \ldots, p(x_j|\acute{x}_t)\right) \tag{3.8}$$

A each iteration, the patch with highest saliency is selected. This sequential approach selects patches that give the highest average increase in maximum predicted probability for the patches in the image. As a result, patches that convey similar information as those already chosen are unlikely to be selected.

3.5 Existing Saliency Measures

In our experiments, we compare our proposed contextual saliency measures with three classes of existing saliency measures. The first baseline measure is the uniform saliency measure across the entire image, where patches are sampled randomly from the image. Equivalently, this can be thought of as computing a distribution over the codewords using a normalized histogram. The distribution over codewords is then randomly sampled. The second measure is an interest-point based saliency measure. More specifically, we apply the Harris corner detector [39] to the image, and used its response at every location in the image as the saliency map. We also provide experiments using the patches found from the SIFT detector [38]. Finally, we compare against a discriminative saliency measure. The discriminative measure considers a patch to be salient if the mutual information of the patch and the class labels is high. More specifically, if $\mathcal{M}(w_a)$ is the mutual information of the a^{th} word with the class labels, the measure is defined as $\mathcal{S}_i^D = \sum_{a=1}^{m} \alpha_{ia} \mathcal{M}(w_a)$.

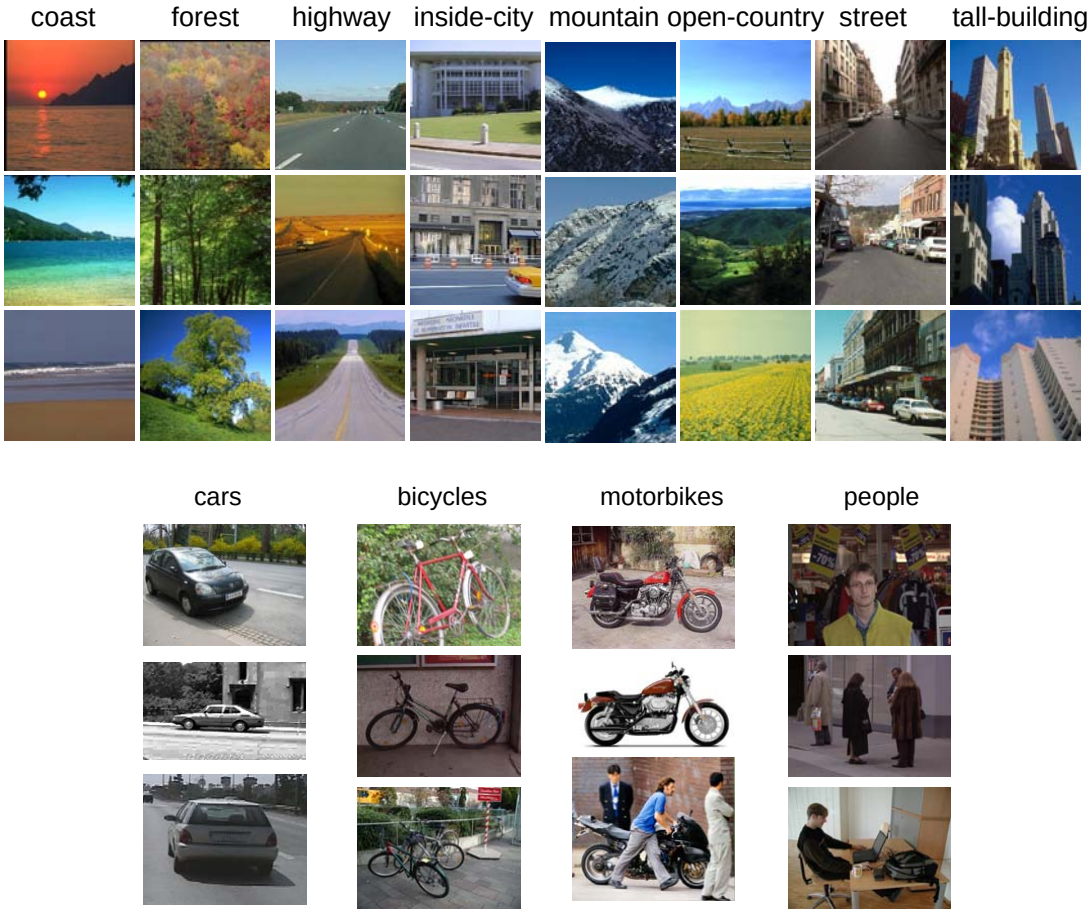


Figure 3.1: Example images from the (top) outdoor scene category dataset [1] and (bottom) Pascal-01 object recognition dataset [2].

3.6 Experimental Setup

We evaluate our proposed contextual saliency measure for the tasks of scene and object recognition using the bag-of-features approach. In both scenarios we construct a codebook of feature descriptors using the standard K-means clustering technique with $K = 1000$. Classification is accomplished by first assigning each sampled patch to a codeword. A histogram of codewords is created as input into a Support Vector Machine (SVM) classifier. We use a Gaussian kernel SVM in all

our experiments. We also experimented with adaptively thresholded histograms by picking thresholds that maximize mutual information with the class labels as suggested by Nowak *et al.* [44]. However, the results were comparable, so we only report experiments using the normalized histograms in our experiments. Experiments using the nearest-neighbor classifier were also tested. The results were consistently inferior, and hence are not included here.

3.6.1 Scene Recognition

We evaluate our approaches on the outdoor scene category dataset from Torralba *et al.* [1]. Example images from this dataset are shown in Fig. 3.1 (top). It contains images from 8 categories: coast, mountain, forest, open country, street, inside city, tall buildings and highways. There are a total of 2866 256×256 color-images. For scene recognition our 48 dimensional descriptor consists of the average color values sampled in a 4×4 grid. The patches were sampled evenly across the image on a 64×64 grid. The patch scale was set so that neighboring patches overlap by 75%. Each sampled patch is given a soft assignment to the codewords using equation (3.3) with $\sigma_w = 30$. Similar to Torralba *et al.* [70], we use 100 images per scene category for training, and the rest as testing.

3.6.2 Object Recognition

Our experiments on object recognition use the Pascal-01 [2] dataset which contains 4 object categories: cars, bicycles, motorbikes and people. Example images from the dataset are shown in Fig. 3.1 (bottom). A training set of 684 images and a test set of 689 images is defined. Since object recognition is more dependent on

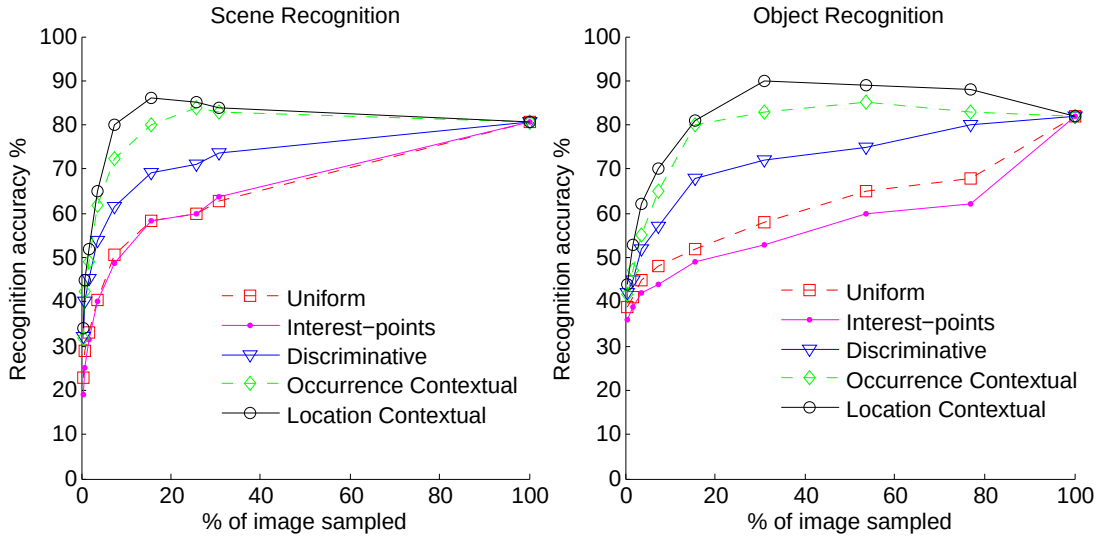


Figure 3.2: Scene (left) and object (right) recognition accuracies for different saliency measures. The weighted random sampling strategy is used in all cases.

image gradients than color, we use SIFT [38] as our descriptor. The descriptor was sampled on a 64×64 uniformly spaced grid. The scale of the sampled patches was set so that horizontally neighboring patches overlap by 75%. In this scenario we used hard assignments of patches to codewords.

3.7 Results

3.7.1 Comparing Saliency Measures

Our first experiments test our contextual saliency measures and those described in Section 3.5 on the scene and object recognition datasets. Recognition accuracies are plotted relative to the density of samples used for classification in Fig. 3.2. The weighted random sampling strategy is used in all cases. For comparison, representative reported accuracies on these datasets are 84% on the outdoor scene

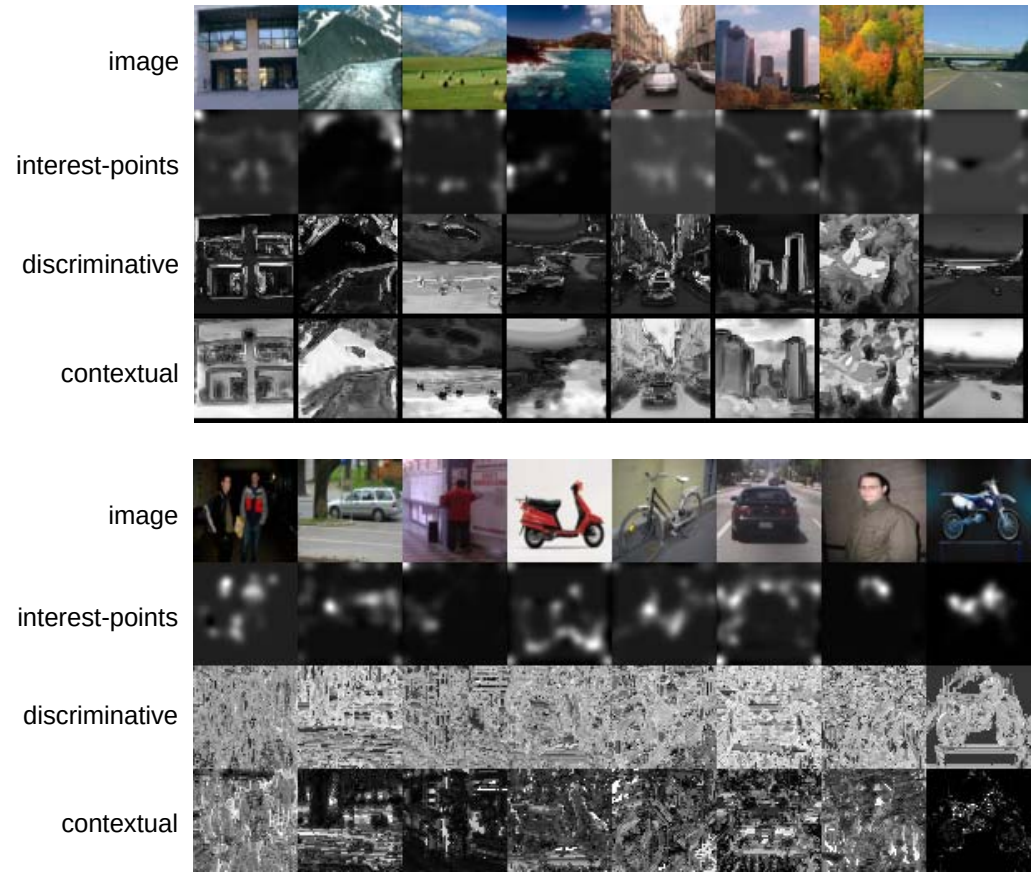


Figure 3.3: Example saliency maps for images for the (top) scene recognition and (bottom) object recognition tasks using different classes of saliency measures. Maps are normalized to lie between 0 (least salient patch) and 1 (most salient patch)

recognition dataset by Torralba *et al.* [70] and $\sim$ 88% on the Pascal-01 object recognition dataset by Nowak *et al.* [44]. The highest accuracies achieved by the contextual saliency measures are 84% and 86% on the scene recognition task for $\mathcal{S}^l$ and $\mathcal{S}^o$ respectively, and 85% and 90% for the object recognition task. We also tested our algorithm using higher resolution grids for scene recognition. The highest accuracies for $\mathcal{S}^o$ were 55%, 68% and 81% for 8×8, 16×16 and 32×32 sampled grids respectively.

The contextual saliency measures have the best performance on both datasets. The discriminative measure using mutual information is next followed by similar results for both interest points and random saliency measures. We also ran experiments using the complete set of interest points found using the SIFT detector. On average the SIFT detector found 926 interest points and a recognition accuracy of 71% was achieved on the object recognition dataset. The performance of the saliency measures not using context increase monotonically with the density of the sampling. This is consistent with observations made by Nowak *et al.* [44]. However, with the contextual saliency measures a sparser sampling results in higher accuracy. We believe this is due to the presence of noisy or irrelevant patches in the image. Our saliency measure can pick out the relevant/representative patches in the image first, but if we keep adding more patches, we incorporate noise in the data, making the classification task harder. This indicates that a sparser sampling is desirable not only for computational efficiency, but also higher recognition performance. With respect to the saliency measures, the usefulness of spatial information varies based on the dataset. The spatial information provides a larger performance boost for object recognition. We speculate that this is due to the increased spatial ambiguity of SIFT descriptors as compared to color descriptors. In scene recognition, color descriptors such as blue patches that correspond to sky or green patches that correspond to grass are strongly correlated with certain image locations. As a result, the spatial information may be redundant.

The various saliency maps for a set of sample images are shown in Fig. 3.3. In the scene recognition examples, objects that are unlikely given the scene category typically have lower saliency measures. As can be seen in Fig. 3.3(b), the saliency maps for the object recognition datasets have high saliency values even for the

backgrounds. We believe this is due to several reasons: a strong correlation of objects and background, the higher entropy of the SIFT descriptor and the use of hard assignments.

Figure 3.4: Red patches on the car in the highway image (left) and the white patches from the light behind the trees in the forest image (right) are considered to be salient by the discriminative measure because they occur pre-dominantly in sunset coast and snow-covered mountain images respectively. However, the contextual saliency measure incorporates the context of the rest of the scene and thus considers the road, sky and tress to be salient instead.

The higher performance of contextual saliency measures over discriminative saliency measures may seem un-intuitive at first, since the discriminative saliency measure is supervised and is optimized specifically for recognition accuracies. However, it should be noted that the discriminative saliency measure ignores the rest of the scene, or the context in which the patch is present. This can lead to undesirable artifacts. For instance, in a scene recognition task, red/orange patches may be considered to be salient by the discriminative measure since they occur pre-dominantly only in sunset (coast) images. Consider a highway test image that has a red car present in it, as seen in Fig. 3.4 (left). All the patches on the car will be considered highly salient by the discriminative saliency measure even though they are not representative of the scene. The saliency measure using context would identify that the red patches on the car are not representative of the image, and would not use them for classification. As a result, the contextual saliency measure is more likely to ignore clutter in the scene, resulting in higher

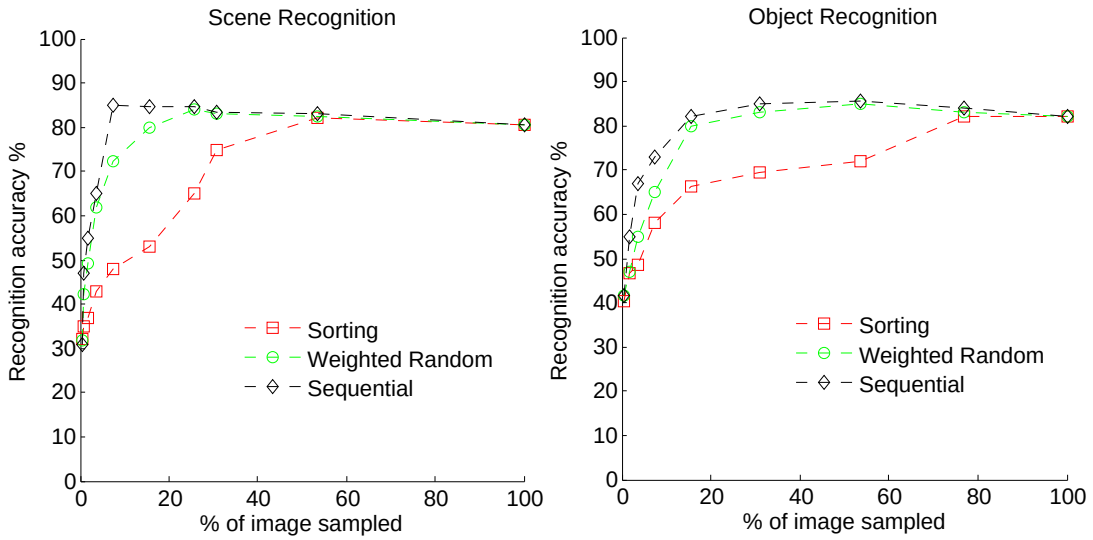


Figure 3.5: Scene (left) and object (right) recognition accuracies for different sampling strategies. The occurrence-based contextual saliency measure $\mathcal{S}^o$ is used in all cases.

accuracies. A similar result can be seen in Fig. 3.4 (right).

3.7.2 Comparing Sampling Strategies

To compare the different sampling strategies described in Section 3.4, we work with the occurrence-based contextual saliency measure. The scene and object recognition accuracies using the different sampling strategies are shown in Fig. 3.5. We can see that for scene recognition, the sorting strategy is much worse than the weighted random sampling. The features used for the scene recognition task are raw color patches, and hence neighboring patches in an image have very similar features and hence very similar saliency measures. While sequential sampling does not give higher accuracies, it reaches the peak accuracy using fewer patches than the weighted random sampling. The reason for the sequential sampling not outperforming the weighted sampling may be that both methods only pick out

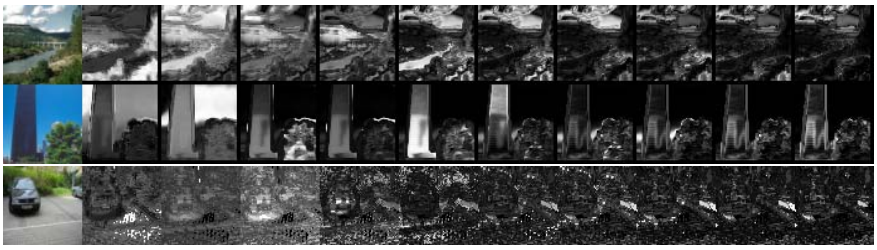


Figure 3.6: Illustration of sequential sampling. Left: original image; Subsequent columns: saliency map being updated at each iteration; Top two rows: scene recognition; Bottom row: object recognition.

the relevant patches, and the drawback of the weighted sampling strategy is only that it often picks out redundant patches (which are also relevant for the task, but only redundant in the presence of other patches already sampled). Similar trends are seen for object recognition with the sequential and weighted random sampling results being more comparable. This may be due to the lower correlation of neighboring SIFT features as compared to the color descriptors used for scene recognition. Examples of how the saliency maps are updated after each iteration of the sequential sampling are shown in Fig. 3.6.

3.7.3 Discussion

Typically, contextual information is used for high-level reasoning about interactions between objects. In this chapter, we demonstrate how contextual information may also be useful for low-level applications such as measuring patch saliency. While low-level contextual reasoning lacks semantic object information, even color or texture patches can supply useful contextual information as also shown in [13].

Discriminative saliency measures capture classification specific statistics of the

patches. Our proposed contextual saliency measures capture contextual information of the entire image to determine saliency of patches. Both these aspects are complementary, and are both important to select representative patches that give good recognition accuracies. For the data sets we experimented with so far, the contextual information was more critical than the discriminative information, such as the example shown in Figure 3.4. However, one can imagine scenarios where the reverse is true. The balance between the two is task and domain dependent. A natural future direction, hence, is to combine discriminative and contextual information to design the optimum saliency measure for a given task. This is related to subjectiveness in the notion of saliency itself. Salient regions may be considered to be those that are representative of the image (as we do in this work), or those that are rare or unusual and hence draw attention. If we consider a more generic definition of saliency as being informative, it leads us back to the notion of task and domain dependency.

While our contextual saliency measure is unsupervised it is still dataset specific. That is, training images are needed to learn the co-occurrence statistics of the codewords. Other methods such as the use of interest points or random sampling may be better suited for applications in which the statistics of the images may not be known beforehand.

3.8 Conclusions

In this chapter we propose two measures of saliency using contextual information. The first measure relies on co-occurrence information between codewords, while the second measure includes spatial information. We test our saliency measures

against several others using the bag-of-features paradigm. Our experiments show improved results over other saliency measures on both scene and object recognition datasets. In contrast to previous works that produce results with accuracies that monotonically increase with sampling density, the contextual saliency measures produce optimal results with a sparse sampling. We demonstrate the use of contextual information for a low-level task, as opposed to the traditional high-level tasks.

Chapter 4

How: Unsupervised Modeling of Objects and their Hierarchical Contextual Interactions

Summary

A successful representation of objects in literature is as a collection of patches, or parts, with a certain appearance and position. The relative locations of the different parts of an object are constrained by the geometry of the object. Going beyond a single object, consider a collection of images of a particular scene category containing multiple (recurring) objects. The parts belonging to different objects are not constrained by such a geometry. However the objects themselves, arguably due to their semantic relationships, demonstrate a pattern in their relative locations. Hence, analyzing the interactions among the parts across the collection of images can allow for extraction of the foreground objects,

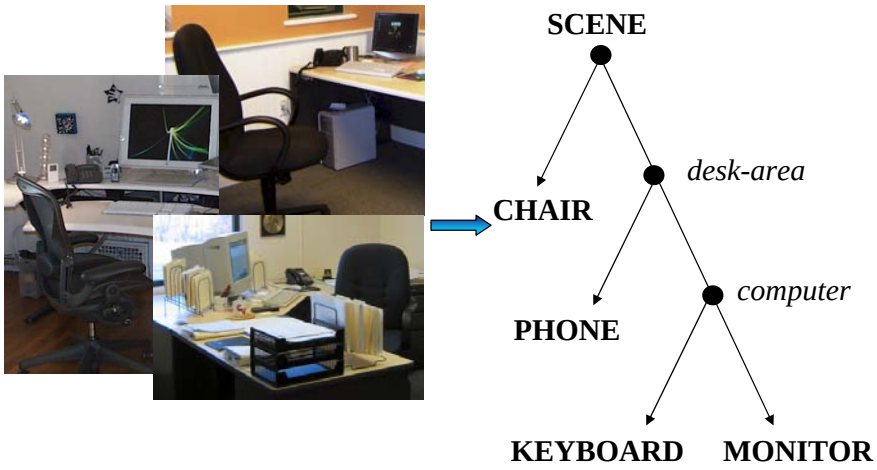


Figure 4.1: Images for "office" scene from Google image search. There are four commonly occurring objects: chair, phone, monitor and keyboard. The monitor and keyboard occur at similar relative locations across images and hence belong to a common super-object, computer, at a lower level in the hierarchy. The phone is seen within the vicinity of the monitor and keyboard. However, the chair is arbitrarily placed, and hence belongs to a common super-object with other objects only at the highest level in the hierarchy, the entire scene. This pattern in relative locations, often stemming from semantic relationships among the objects, provides contextual information about the scene "office" and is captured by an hSO: Hierarchical Semantics of Objects. A possible corresponding hSO is shown on the right.

and analyzing the interactions among these objects can allow for a semantically meaningful grouping of these objects that characterizes the entire scene. These groupings are typically hierarchical. We introduce hSO: Hierarchical Semantics of Objects, that captures this hierarchical grouping. We propose an approach for the unsupervised learning of the hSO from a collection of images of a particular scene. We also demonstrate the use of the hSO in providing context for enhanced object localization in the presence of significant occlusions, and show its superior performance over a fully connected graphical model for the same task.

4.1 Introduction

Objects that tend to co-occur in scenes are often semantically related. Hence, they demonstrate a characteristic grouping behavior according to their relative positions in the scene. Some groupings are tighter than others, and thus a hierarchy of these groupings among these objects can be observed in a collection of images of similar scenes. It is this hierarchy that we refer to as the Hierarchical Semantics of Objects (hSO). This can be better understood with an example.

Consider an office scene. Most offices, as seen in Figure 4.1, are likely to have, for instance, a chair, a phone, a monitor and a keyboard. If we analyze a collection of images taken from such office settings, we would observe that across images, the monitor and keyboard are more or less in the same position with respect to each other, and hence can be considered to be part of the same super-object at a lower level in the hSO structure - say a computer. Similarly, the computer may usually be somewhere in the vicinity of the phone, and so the computer and the phone belong to the same super-object at a higher level - say the desk-area. But the chair and the desk-area may be placed relatively arbitrarily in the scene with respect to each other, more so than any of the other objects, and hence belong to a common super-object only at the highest level in the hierarchy i.e. the scene itself. A possible hSO that would describe such an office scene is shown in Figure 4.1. Along with the structure, the hSO may also store other information such as the relative position of the objects and their co-occurrence counts as parameters.

The hSO is motivated from an interesting thought exercise: at what scale is an object defined? Are the individual keys on a keyboard objects, or the entire

keyboard, or is the entire computer an object? The definition of an object is blurry, and the hSO exploits this to allow incorporation of semantic information of the scene layout. The leaves of the hSO are a collection of parts and represent the objects, while the various levels in the hSO represent the super-objects at different levels of abstractness, with the entire scene at the highest level. Hence hSOs span the spectrum between specific objects, modeled as a collection of parts, at the lower level and scene categories at the higher level. This provides a rich amount of information at various semantic levels that can be potentially exploited for a variety of applications, ranging from establishing correspondences between parts for object matching and providing context for robust object detection, all the way to scene category classification.

In most works that attempt to learn such relationships among objects, the machine is trained with excessive guidance or supervision. This is in stark contrast with how humans, in my opinion, seem to develop an understanding for the visual world that surrounds us.

We grasp very rich information, such as relationships among objects or parts of objects or actions or events, just by observing and experiencing the visual world around us. For instance, is one ever explicitly told that a monitor is closely related to a keyboard? Or that when one person takes a swing at someone they duck? Or that a motorbike (almost) always has two wheels and a handle-bar? We pick these interactions up without having to memorize a handbook defining such relationships. One may argue that cues such as functionality of objects, or even evolutionary instincts, aid us in understanding these interactions. This may certainly be true to an extent, however I believe that visual cues, such as the relative locations of objects, detachability of parts of objects, temporal correlations

among actions and events in video, and others play a very fundamental role in establishing this understanding.

Indeed, it seems very plausible that from a collection of office images, without any annotations, the machine can learn that keyboards always occur at consistent relative locations with respect to the monitors. Similarly, from a collection of street images, the machine can learn that cars are always on the road, in front of building facades, the building facades often have windows on them, cars have wheels, and so on. It should be noted that since we are considering an image collection with no annotations, the algorithm would not know the names such as car and road but can have the visual patterns in its knowledge that correspond to these words. Similarly, from a collection of surveillance videos the machine can learn that usually a person walks across the parking lot to his car, places his briefcase down, unlocks the car, opens the door, and puts the briefcase inside before driving away.

These learnt interactions may be applied for enhancing several computer vision tasks such as object recognition, object detection, or when applied to video, for action and event recognition, anomaly detection, and others. For example, if the robot has automatically learnt that monitors and keyboards are strongly associated with each other in an office scene, it can perform more robust keyboard detection by incorporating the context provided by the location of the monitor. Having observed the parking lot videos, it can learn that if someone leaves their briefcase behind and drives away, it is an anomaly.

Scenes may contain several objects of interest, and hand labeling these objects would be quite tedious. To avoid this, as well as the bias introduced by the subjectiveness of a human in identifying the objects of interest in a scene, unsu-

pervised learning of hSO is preferred so that it truly captures the characteristics of the data. Also, unsupervised learning ensures that the true underlying interactions of our world are learnt without any biases introduced by human labeling. And perhaps most importantly, unsupervised algorithms are often general enough to be applied to a wide variety of domains/scenarios/applications ranging from indoor scene understanding to surveillance and perhaps medical imaging as opposed to being geared towards a specific task at hand. The same robot, running the same algorithm, can be used in the office setting to understand interactions among objects such as monitors and keyboards, or in the surveillance setting to understanding interactions among temporal actions such as walking and events such as theft.

In this chapter we introduce Hierarchical Semantics of Objects (hSO). We propose an approach for unsupervised learning of hSO from a collection of images. This algorithm is able to identify the foreground parts in the images, cluster them into objects and further cluster the objects into a hierarchical structure that captures semantic relationships among these objects - all in an unsupervised (or semi-supervised, considering that the images are all from a particular scene) manner from a collection of unlabeled images. We demonstrate the superiority of our approach for extracting multiple foreground objects as compared to some benchmarks. Furthermore, we also demonstrate the use of the learnt hSO in providing object models for object localization, as well as context to significantly aid localization in the presence of occlusion. We show that an hSO is more effective for this task than a fully connected network.

The following two chapters in this thesis present our approach to unsupervised learning of hierarchical spatial patterns in images. In this chapter, we present an

approach geared towards images taken of the same scene over a period of time, hence containing the same foreground object instances. We specifically leverage the fact that our scene contains the same object instances, and hence explicitly exploit the geometry constraints within these objects. The resultant hierarchies capture interactions among objects, and sit well with our intuitions and semantic understanding of the scene. In the following chapter we present an approach that can deal with a generic collection of images of object categories, scene categories, etc. The resultant hierarchies capture statistical interactions among low-level features, parts of objects, objects and groups of objects. These entities are modeled statistically, and may not correlate completely with semantics. Both these approaches have their merits, and would be appropriate depending on the application and scenario at hand.

The rest of this chapter is organized as follows. Section 4.2 describes related work in literature. Section 4.3 describes some applications that motivate the need for hSO, and discusses prior works for these applications as well. Section 4.4 describes our approach for the unsupervised learning of hSO from a collection of images. Section 4.5 presents our experimental results in identifying the foreground objects and learning the hSO. Section 4.6 presents our approach for utilizing the information in the learnt hso as context for object localization, followed by experimental results for the same. Section 4.7 concludes the chapter.

4.2 Related Work

Different aspects of this work have appeared in [20, 71]. We modify the approach presented in [20] by adopting techniques presented in [71]. Moreover, we pro-

pose a formal approach for utilizing the information in the learnt hso as context for object localization. We present thorough experimental results for this task including quantitative analysis and compare the accuracies of our proposed hierarchy (tree-structure) among objects to a flat fully connected model/structure over the objects.

4.2.1 Foreground identification

The first step in learning the hSO is to first extract the foreground objects from the collection of images of a scene. In our approach we focus on rigid objects. We exploit two intuitive notions to extract the objects. First, the parts of the images that occur frequently across images are likely to belong to the foreground. And second, only those parts of the foreground that are found at geometrically consistent relative locations are likely to belong to the same rigid object.

Several approaches in literature address the problem of foreground identification. First of, we differentiate our approach for this task from image segmentation approaches. These approaches are based on low level cues and aim to separate a given image into several regions with pixel level accuracies. Our goal is a higher level task, where using cues from multiple images, we wish to separate the local parts of the images that belong to the objects of interest from those that lie on the background. To re-iterate, several image segmentation approaches aim at finding regions that are consistent within a single image in color, texture, etc. We are however interested in finding objects in the scene that are consistent across multiple images in occurrence and geometry.

Several approaches for discovering the *topic* of interest have been proposed

such as discovering main characters [72] or objects and scenes [73] in movies or celebrities in collections of news clippings [74]. Recently, statistical text analysis tools such as probabilistic Latent Semantic Analysis (pLSA) [75] and Latent Dirichlet Allocation (LDA) [76] have been applied to images for discovering object and scene categories [61, 77, 56]. These use unordered *bag-of-words* [52] representation of documents to automatically (unsupervised) discover topics in a large corpus of documents/images. However these approaches, which we loosely refer to as *popularity* based approaches, do not incorporate any spatial information. Hence, while they can identify the foreground from the background, they can not further separate the foreground into multiple objects. Hence, these methods have been applied to images that contain only one foreground object. We further illustrate this point in our results. These popularity based approaches can separate the multiple objects of interest only if they are provided images that contain different number of these objects. For the office setting, in order to discover the monitor and keyboard separately, pLSA, for instance, would require several images with just the monitor, and just the keyboard (and also a specified number of topics of interest). This is not a natural setting for images of office scenes. Leordeanu, *et al.* [78] propose an approach for the unsupervised learning of the object model from its low resolution video. However, this approach is also based on co-occurrence and hence can not separate out multiple objects in the foreground.

Several approaches have been proposed to incorporate spatial information in the popularity based approaches [79, 80, 54, 81], however, only with the purpose of robustly identifying the single foreground object in the image, and not for separation of the foreground into multiple objects. Russell, *et al.* [82], through their

approach of breaking an image down into multiple segments and treating each segment individually, can deal with multiple objects as a byproduct. However, they rely on consistent segmentations of the foreground objects, and attempt to obtain those through multiple segmentations.

On the object detection/recognition front, approaches such as applying object localization classifiers through a sliding window approach, could be considered, with a stretch of argument, to provide rough foreground/background separation. However, these are supervised methods. Part-based approaches, like ours, however towards this goal of object localization, have been proposed such as [83, 84] which use spatial statistics of parts to obtain objects masks. These are supervised approaches as well, and for single objects. Unsupervised part-based approaches for learning the object models for recognition have also been proposed, such as [3, 85]. These also deal with single objects.

4.2.2 Modeling dependencies among parts

Several approaches in text data-mining represent the words in a lower dimensional space where words with supposedly similar semantic meanings collapse into the same cluster. This representation is based simply on their occurrence counts in documents. pLSA [75] is one such approach that has also been applied to images [56, 61, 77] for unsupervised clustering of images based on their *topic* and identifying the part of the images that are foreground. Our goal however is a step beyond this towards a higher level understanding of the scene. Apart from simply identifying the *existence* of potential semantic relationships between the parts, we attempt to characterize these semantic relationships, and accordingly

cluster the parts into (super) objects at various levels in the hSO. Several works [86, 87] model dependencies among parts of a single object for improved object recognition/detection. Our goal however is to model correlations among multiple objects and their parts. We define dependencies based on relative location as opposed to co-occurrence.

It is important to note that, our approach being entirely unsupervised, the presence of multiple objects as well as background clutter makes the task of clustering the foreground parts into hierarchial clusters, while still maintaining the integrity of objects yet capturing the inter-relationships among them, challenging. The information coded in the learnt hSO is hence quite rich. It entails more than a mere extension of the above works to multiple objects.

4.2.3 Hierarchies

Using hierarchies or dependencies among parts of objects for object recognition has been promoted for decades [88, 89, 90, 91, 92, 86, 87, 93, 94]. However we differentiate our work from these, as our goal is not object recognition, but is to characterize the scene by modeling the interactions between multiple objects in a scene. More so, although these works deal with hierarchies per say, they capture philosophically very different phenomena through the hierarchy. For instance, Marr *et al.* [88] and Levinshtein *et al.* [91] capture the shape of articulated objects such as the human body through a hierarchy, where as Fidler *et al.* [94] capture varying levels of complexity of features. Bienenstock *et al.* [90] and Siskind *et al.* [95] learn a hierarchical structure among different parts/regions of an image based on rules on absolute locations of the regions in the images, similar

to those that govern the grammar or syntax of a language. These various notions of hierarchy are strikingly different from the inter-object, potentially semantic, relationships that we wish to capture through a hierarchical structure.

4.3 Applications of hSO

Before we describe the details of the learning algorithm, we first motivate hSOs through a couple of interesting potential areas for their application.

4.3.1 Context

Learning the hSO of scene categories could provide contextual information for tasks such as object recognition, detection or localization. The accuracy of individual detectors can be enhanced as the hSO provides a prior over the likely position of an object, given the position of another object in the scene.

Consider the example shown in Figure 4.1. Suppose we have independent detectors for monitors and keyboards. Consider a particular test image in which a monitor is detected. However there is little evidence indicating the presence of a keyboard - due to occlusion, severe pose change, etc. The learnt hSO (with parameters) for office settings would provide the contextual information indicating the presence of a keyboard and also an estimate of its likely position in the image. If the observed bit of evidence in that region of the image supports this hypothesis, a keyboard may be detected. However, if the observed evidence is to the contrary, not only is the keyboard not detected, but the confidence in the detection of the monitor is reduced as well. The hSO thus allows for propagation of such information among the independent detectors.

Several works use context for better image understanding. One class of approaches involves analyzing individual images for characteristics of the surroundings of the object such as geometric consistency of object hypotheses [96], viewpoint and mean scene depth estimation [10, 97], surface orientations [8], etc. These provide useful information to enhance object detection/recognition. However, our goal is not to extract information about the surroundings of the object of interest from a single image. Instead, we aim to learn a characteristic representation of the scene category and a more higher level understanding from a collection of images by capturing the semantic interplay among the objects in the scene as demonstrated across the images.

The other class of approaches models dependencies among different parts of an image [98, 99, 100, 11, 12, 16, 101] from a collection of images. However, these approaches require hand annotated or labeled images. Also, [98, 99, 100, 12] are interested in pixel labels (image segmentation) and hence do not deal with the notion of *objects*. Torralba *et al.* [19] use the global statistics of the image to predict the type of scene which provides context for the location of the object, however their approach is also supervised. Torralba *et al.* [9] learn interactions among the objects in a scene for context, however their approach is supervised and the different objects in the images need to be annotated. Marszałek *et al.* [102] also learn relationships among multiple classes of objects, however indirectly through a lexical model learnt on the labels given to images, and hence is a supervised approach. Our approach, is entirely unsupervised - the relevant parts of the images, and their relationships are automatically *discovered* from a corpus of unlabeled images.

4.3.2 Compact scene category representation

hSOs provide a compact representation that characterizes the scene category of the images that it has been learnt from. Hence, hSOs can be used for scene category classification. Singhal *et al.* [17] learn a set of relationships between different regions in a large collection of images with a goal to characterize the scene category. However, these images are hand segmented, and a set of possible relationships between the different regions are predefined (above, below, etc.). Other works [1, 103] also categorize scenes but require extensive human labeling. Fei-Fei *et al.* [61] group the low-level features into *themes* and *themes* into scene categories. However, the *themes* need not corresponding to semantically meaningful entities. Also, they do not include any location information, and hence cannot capture the interactions between different parts of the image. They are able to learn an hierarchy that relates the different scenes according to their similarity, however, our goal is to learn an hierarchy for a particular scene that characterizes the interactions among the entities in the scene, arguably according to the underlying semantics.

4.3.3 Anomaly detection

As stated earlier, the hSO characterizes a particular scene. It goes beyond an occurrence based description, and explicitly models the interactions among the different objects through their relative locations. Hence, it is capable of distinguishing between scenes that contain the same objects, however in different configurations. This can be useful for anomaly detection. For instance, consider the office scene in Figure 4.1. In an office input image, if we find the objects at

locations in very unlikely configurations given the learnt hSO, we can detect a possible intrusion in the office or some such anomaly.

These examples of possible applications for the hSO demonstrate its use for object level tasks such as object localization, scene level tasks such as scene categorization and one that is somewhere in between the two: anomaly detection. Later in this chapter we demonstrate the use of hSO for the task of robust object localization in the presence of occlusions.

4.4 Unsupervised Learning of hSO

Our approach for the unsupervised learning of hSOs is summarized in Figure 4.2. The input is a collection of images taken in a particular scene, and the desired output is the hSO. The general approach is to first separate the features in the input images into foreground and background features, followed by clustering of the foreground features into the multiple foreground objects, and finally extracting the hSO characterizing the interactions among these objects. Each of the processing stages is explained in detail next.

4.4.1 Feature extraction

Given the collection of images taken from a particular scene, local features describing interest points/parts are extracted in all the images. These features may be appearance based features such as SIFT [38], shape based features such as shape context [104], geometric blur [105], or any such discriminative local descriptors as may be suitable for the objects under consideration. In our current implementation, we use the Difference of Gaussian interest point detector, and

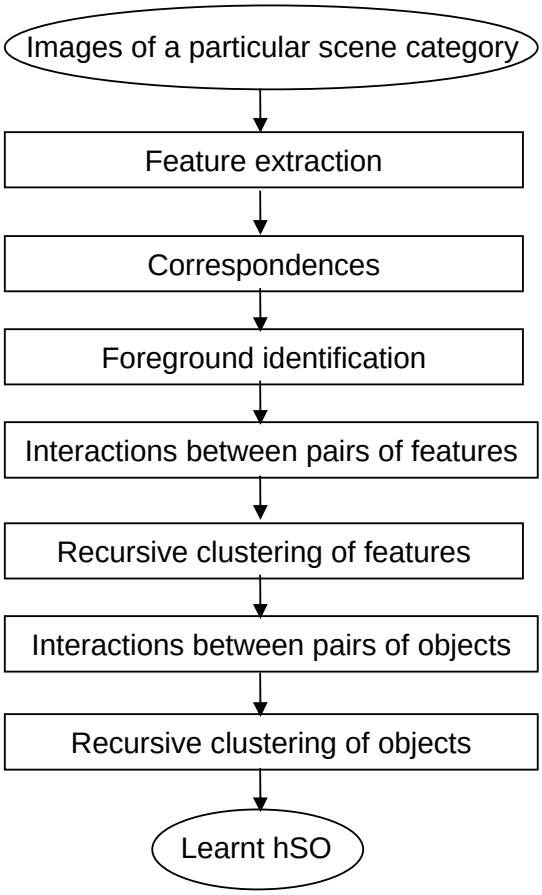


Figure 4.2: Flow of the proposed algorithm for the unsupervised learning of hSOs

SIFT features as our local descriptors.

4.4.2 Correspondences

Having extracted features from all images, correspondences between these local parts are identified across images. For a given pair of images, potential correspondences are identified by finding k nearest neighbors of each feature point from one image in the other image. We use Euclidean distance between the SIFT descriptors to determine the nearest neighbors. The geometric consistency between every pair of correspondences is computed to build a geometric consistency

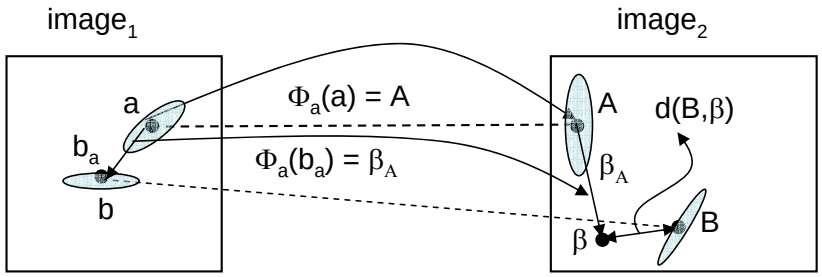


Figure 4.3: An illustration of the geometric consistency metric used to retain *good* correspondences.

adjacency matrix.

Suppose we wish to compute the geometric consistency between a pair of correspondences shown in Figure 4.3 involving interest regions a and b in $image_1$ and A and B in $image_2$. All interest regions have a scale and orientation associated with them. Let ϕ_a be the similarity transform that transforms a to A. β_A is the result of the transformation of b_a (the relative location of b with respect to a in $image_1$) under ϕ_a. β is thus the estimated location of B in the $image_2$ based on ϕ_a. If a and A, as well as b and B are geometrically consistent, the distance between β and B, $d(B, \beta)$ would be small. A score that decreases exponentially with increasing $d(B, \beta)$ is used to quantify the geometric consistency of the pair of correspondences. To make the score symmetric, a is similarly mapped to α under the transform ϕ_b that maps b to B, and the score is based on $max(d(B, \beta), d(A, \alpha))$. This metric provides us with invariance only to scale and rotation, the assumption being that the distortion due to affine transformation in realistic scenarios is minimal among local features that are closely located on the same object.

Having computed the geometric consistency score between all possible pairs of correspondences, a spectral technique is applied to the geometric consistency ad-

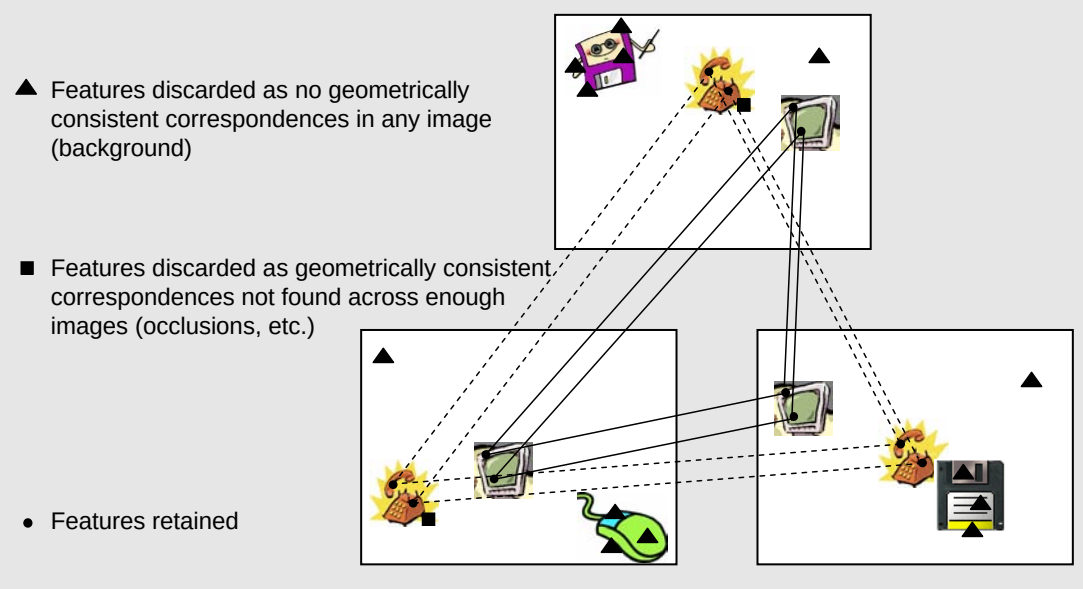


Figure 4.4: An illustration of the correspondences and features retained. For clarity, the images contain only two of the four foreground objects we have been considering in the office scene example from Figure 4.1, and some background.

jacency matrix to retain only the geometrically consistent correspondences [106]. This helps eliminate most of the background clutter. This also enables us to deal with incorrect low-level correspondences among the SIFT features that can not be reliably matched, for instance at various corners and edges found in an office setting. To deal with multiple objects in the scene, an iterative form of [106] is used. However, it should be noted that due to noise, affine and perspective transformations of objects, etc. correspondences of all parts even on a single object do not always form one strong cluster and hence are not entirely obtained in a single iteration, instead they are obtained over several iterations.

4.4.3 Foreground identification

Only the feature points that find geometrically consistent correspondences in most other images are retained. This is in accordance with our perception that the objects of interest occur frequently across the image collection. Also, this post processing step helps to eliminate the remaining background features that may have found geometrically consistent correspondences in another image by chance. Using multiple images gives us the power to be able to eliminate these random errors which would not be consistent across images. However, we do not require features to be present in all images in order to be retained. This allows us to handle occlusions, severe view point changes, etc. Since these affect different parts of the objects across images, it is unlikely that a significant portion of the object will not be matched in many images, and hence be eliminated by this step. Also, this enables us to deal with different number of objects in the scene across images, the assumption being that the objects that are present in most images are the objects of interest (foreground), while those that are present in a few images are part of the background clutter. This proportion can be varied to suit the scenario at hand.

We now have a reliable set of *foreground* feature points and a set of correspondences among all images. An illustration can be seen in Figure 4.4 where only a subset of the detected features and their correspondences are retained. It should be noted that the approach being unsupervised, there is no notion of an object yet. We only have a cloud of features in each image which have all been identified as foreground and correspondences among them. The goal is to now separate these features into different groups, where each group corresponds to a

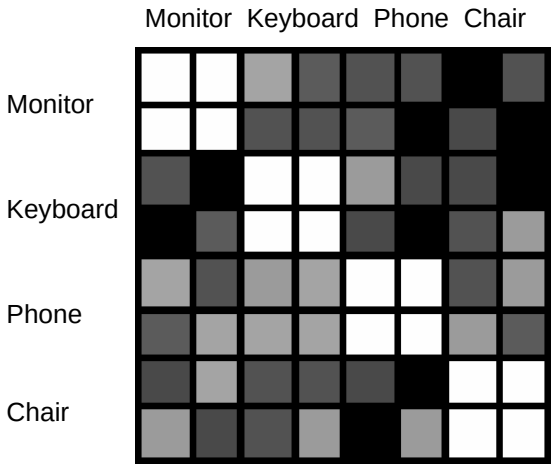


Figure 4.5: An illustration of the geometric consistency adjacency matrix of the graph that would be built on all retained foreground features for the office scene example as in Figure 4.1.

foreground object in the scene, and further learn the hierarchy among these objects that will be represented as an hSO that will characterize the entire collection of images and hence the scene.

4.4.4 Interaction between pairs of features

In order to separate the cloud of retained feature points into clusters, a graph is built over the feature points, where the weights on the edge between the nodes represents the interaction between the pair of features across the images. The metric used to capture the interaction between the pairs of features is the same geometric consistency as computed in Section 4.4.2, except now averaged across all pairs of images that contain these features. While the geometric consistency could contain errors for a particular pair of images due to errors in correspondences, etc. averaging across all pairs suppresses the contribution of these erroneous matchings and amplifies the true interaction among the pairs of features.

If the geometric consistency between two feature points is high, they are likely to belong to the same rigid object. On the other hand, features that belong to different objects would be geometrically inconsistent because the different objects are likely to be found in different configurations across images. An illustration of the geometric consistency adjacency matrix can be seen in Figure 4.4. Again, there is no concept of an object yet. The features in Figure 4.4 are arranged in an order that correspond to the objects, and each object is shown to have only two features, only for illustration purposes.

4.4.5 Recursive clustering of features

Having built the graph capturing the interaction between all pairs of features across images, recursive clustering is performed on this graph. At each step, the graph is clustered into two clusters. The properties of each cluster are analyzed, and one or both of the clusters are further separated into two clusters, and so on. If the variance in the adjacency matrix corresponding to a certain cluster (subgraph) is very low but with a high mean, it is assumed to contain parts from a single object, and is hence not divided further. The approach is fairly insensitive to the thresholds used on the mean and variance of the (sub) adjacency matrix. It can be verified for the example shown in Figure 4.4, that the foreground features would be clustered into four clusters, each cluster corresponding to a foreground object. Since the statistics of each of the clusters formed are analyzed to determine if it should be further clustered or not, the number of foreground objects need not be known *a priori*. This is an advantage as compared to pLSA or parametric methods such as fitting a mixture of Guassians to the foreground

features spatial distribution. Our approach is non-parametric. We use normalized cuts [107] to perform the clustering. The code provided at [108] was used.

4.4.6 Interaction between pairs of objects

Having extracted the foreground objects, the next step is to cluster these objects in a (semantically) meaningful way and extract the underlying hierarchy. In order to do so, a fully connected graph is built over the objects, where the weights on the edges between the nodes represent the interaction between the pairs of objects across the images. The metric used to capture the interaction between the pairs of objects is the predictability of the location of one object, if the location of the other object were known. This is computed as the negative entropy of the distribution of the location of one object conditioned on the location of the other object, or the relative location of one object with respect to the other. The higher the entropy, the less predictable the relative locations are. Let O be the number of foreground objects in our image collection. Suppose $\boldsymbol{M}$ is the $O \times O$ interaction adjacency matrix we wish to create, then $\boldsymbol{M}(i, j)$ holds the interaction between the i^{th} and j^{th} object as

$$\boldsymbol{M}(i, j) = -E[P(l_i - l_j)], \tag{4.1}$$

where, $E[P(x)]$ is the entropy in a distribution $P(x)$, and $P(l_i - l_j)$ is the distribution of the relative location of the i^{th} object with respect to the j^{th} object. In order to compute $P(l_i - l_j)$, we divide the image into an $G \times G$ grid. G was typically set to 10. This can be varied based on the amounts of relative movements the objects demonstrate across images. Across all input images, the

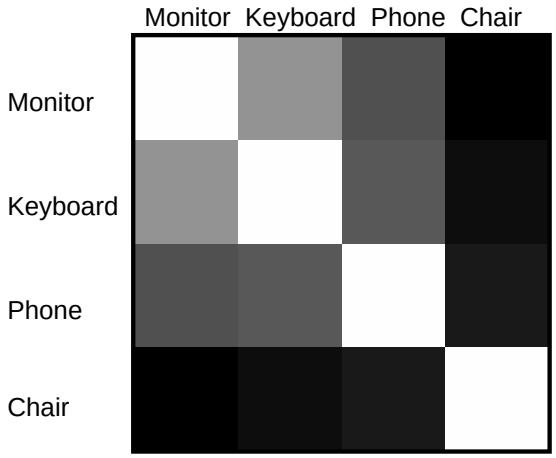


Figure 4.6: An illustration of the entropy based adjacency matrix of the graph that would be built on the foreground objects in the office scene example as in Figure 4.1.

relative locations of the i^{th} object with respect to the j^{th} object are recorded as indexed by one of bins in the grid. We use MLE counts (an histogram like operation) on these relative locations to estimate $P(l_i - l_j)$. If appropriate, the relative locations of objects can be modeled using a Gaussian distribution in which case the covariance matrix would be a direct indicator of the entropy of the distribution. The proposed non-parametric approach is more general. An illustration of the $\boldsymbol{M}$ matrix is shown in Figure 4.6.

4.4.7 Recursive clustering of objects

Having computed the interaction among the pairs of objects, we use recursive clustering on the graph represented by $\boldsymbol{M}$ using normalized cuts. We further cluster every subgraph containing more than one object in it. The objects whose relative locations are most predictable, stay in a common cluster till the end, where as those objects whose locations are not well predicted by most other

objects in the scene are separated out early on. The iteration of clustering at which an object is separated gives us the location of that object in the final hSO. The clustering pattern thus directly maps to the hSO structure. It can be verified for the example shown in Figure 4.6, that the first object to be separated is the chair, followed by the phone and finally the monitor and keyboard, which reflects the hSO shown in Figure 4.1. With this approach, each node in the hierarchy that is not a leaf has exactly two-children. Learning a more general structure of the hierarchy is part of future work.

In addition to learning the structure of the hSO, we also learn the parameters of the hSO. The structure of the hSO indicates that the *siblings* i.e. the objects/super-objects (we refer to them as entities form here on) sharing the same parent node in the hSO structure, are the most informative for each other to predict their location. Hence, during learning, we learn the parameters of the relative location of an entity with respect to its sibling in the hSO only; as compared to learning the interaction among all objects (a flat fully connected network structure instead of hierarchy) where all possible combinations of objects would need to be considered. This would entail learning a larger number of parameters, which for a large number of objects could be prohibitive. Moreover, with limited training images, the relative locations of unrelated objects can not be learnt reliably. This is clearly demonstrated in our experiments in Section 4.6.

The location of an object is considered to be the centroid of the locations of the features that lie on the object. The relative locations are captured non-parametrically as described previously in Section 4.4.6 (parametric estimations could be easily incorporated in our approach). The relative locations of entities in the hSO that are connected by edges are stored (we store the joint distribution

of the location of the two entities and not just the conditional distribution) as MLE counts. The location of a super-object is considered to be the centroid of the locations of the objects composing the super-object. Thus, by storing the relative location of a child with respect to the parent node in the hierarchy, the relative locations of the siblings are indirectly captured. In addition to the relative location statistics, we could also store the co-occurrence statistics.

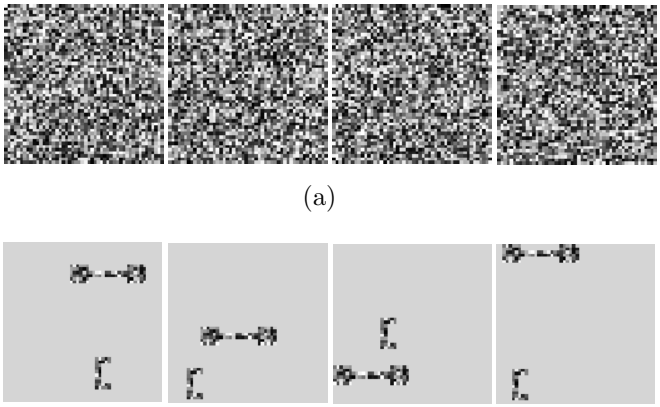
4.5 Experiments

We first present experiments with synthetic images to demonstrate the capabilities of our approach for the subgoal of extracting the multiple foreground objects. The next set of experiments demonstrates the effectiveness of our entire approach for the unsupervised learning of hSO.

4.5.1 Extracting objects

Our approach for extracting the foreground objects of interest uses two aspects: popularity and geometric consistency. These can be loosely thought of as first order as well as second order statistics. In the first set of experiments, we use synthetic images to demonstrate the inadequacy of either of these alone.

To illustrate our point - we consider 50×50 synthetic images as shown in Figure 4.7(a). The images contain 2500 distinct intensity values, of which 128, randomly selected from the 2500, always lie on the foreground objects and the rest is background. We consider each pixel in the image to be an interest point, and the descriptor of each pixel is the intensity value of the pixel. To make visualization clearer, we display only the foreground pixels of these images in



(a)

(b)

Figure 4.7: (a) A subset of the synthetic images used as input to our approach for the unsupervised extraction of foreground objects (b) Background suppressed for visualization purposes.

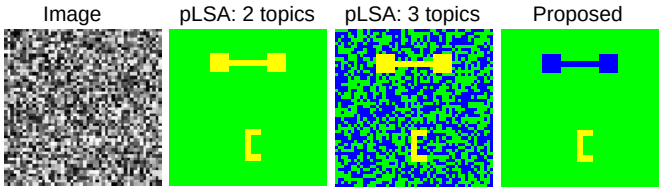


Figure 4.8: Comparison of results obtained using pLSA with those obtained using our proposed approach for the unsupervised extraction of foreground objects.

Figure 4.7(b). It is evident from these that there are two foreground objects of interest. We assume that the objects undergo pure translation only.

We now demonstrate the use of pLSA, as an example of an unsupervised popularity based foreground identification algorithm, on 50 such images. Since pLSA requires negative images without the foreground objects we also provide 50 random negative images to pLSA, which our approach does not need. If we specify pLSA to discover 2 topics, the result obtained is shown in Figure 4.8. It can be seen that it can identify the foreground from the background, but is unable to further separate the foreground into multiple objects. One may argue

Figure 4.9: A subset of images provided as input to learn the corresponding hSO.

that we could further process these results and fit a mixture of Gaussians (for instance) to further separate the foreground into multiple objects. However this would require us to know the number of foreground objects *a priori* and also the distribution of features on the objects need not be Gaussian as in these images. If we specify pLSA to discover 3 topics instead, with the hope that it might separate the foreground into 2 objects, we find that it arbitrarily splits the background into 2 topics, while still maintaining a single foreground topic, as seen in Figure 4.8. This is because pLSA simply incorporates occurrence (popularity) and no spatial information. Hence, pLSA is inherently missing the information required to perceive the features on one of the foreground objects any different than those on the second object, which is required to separate them.

On the other hand, our approach does incorporate this spatial/geometric information and hence can separate the foreground objects. Since the input images are assumed to allow only translation of the foreground objects and the descriptor is simply the intensity value, we alter the notion of geometric consistency than that described in Section 4.4.2. In order to compute the geometric consistency between a pair of correspondences, we compute the distance between the pairs of features in both images. The geometric consistency decreases exponentially as the discrepancy in the distances increases. The result obtained by our ap-

proach is shown in Figure 4.8. We successfully identify the foreground from the background and further separate the foreground into multiple objects. Also, our approach does not require any parameters to be specified, such as number of topics or foreground objects in the images. The inability of a popularity based approach for obtaining the desired results illustrates the need for geometric consistency in addition to popularity.

In order to illustrate the need for considering popularity and not just geometric consistency, let us consider the following analysis. If we consider all pairs of images such as those shown in Figure 4.7 and keep all features that find correspondences that are geometrically consistent with at least one other feature in atleast one other image, we would retain approximately 2300 of the background features. This is because even for background, it is possible to find at least some geometrically consistent correspondences. However the background being random, this would not be consistent across several images. Hence, instead of retaining features that have geometrically consistent correspondences in one other image, if we now retain only those that have geometrically consistent correspondences in at least two other images, only about 50 of the background features are retained. As we use more images, we can eliminate the background features entirely. Our approach being unsupervised, the use of multiple images to prune out background clutter is crucial. Hence, this demonstrates the need for considering popularity in addition to geometric consistency.

4.5.2 Learning hSO

We now present experimental results on the unsupervised learning of hSO from a collection of images. It should be noted that the goal of this work is not improved object recognition through better feature extraction or matching. We focus our efforts at learning the hSO that codes the different interactions among objects in the scene by using well matched parts of objects, and not on the actual matching of parts. This work is complementary to the recent advances in object recognition that enable us to deal with object categories and not just specific objects. These advances indicate the feasibility to learn hSO even among objects categories. However, in our experiments we use specific objects with SIFT features to demonstrate our proposed algorithm. SIFT is not an integral part of our approach. This can easily be replaced with patches, shape features, etc. with appropriate matching techniques as may be appropriate for the scenario at hand - specific objects or object categories. Future work includes experiments in such varied scenarios. Several different experimental scenarios were used to learn the hSOs. Due to lack of standard datasets where interactions between multiple objects can be modeled, we use our own collection of images. The rest of the experiments use the descriptors as well as geometric consistency notions as described in our approach in Section 4.4.

4.5.2.1 Scene semantic analysis

Consider a surveillance type scenario where a camera is monitoring, say an office desk. The camera takes a picture of the desk every few hours. The hSO characterizing this desk, learnt from this collection of images could be used for

robust object detection in this scene, in the presence of occlusion due to a person present, or other extraneous objects on the desk. Also, if the objects on the desk are later found in an arrangement that cannot be explained by the hSO, that can be detected as an anomaly. Thirty images simulating such a scenario were taken. Examples of these can be seen in Figure 4.9. Note the occlusions, background clutter, change in scale and viewpoint, etc. The corresponding hSO as learnt from these images is depicted in Figure 4.10.

Several different interesting observations can be made. First, the background features are mostly eliminated. The features on the right-side of the bag next to the CPU are retained while the rest of the bag is not. This is because due to several occlusions in the images, most of the bag is occluded in images. However, the right-side of the bag resting on the CPU is present in most images, and hence is interpreted to be foreground. The monitor, keyboard, CPU and mug are selected to be the objects of interest (although the mug is absent in some images). The hSO indicates that the mug is found at most unpredictable locations in the image, while the monitor and the keyboard are clustered together till the very last stage in the hSO. This matches our semantic understanding of the scene. Also, since the photo frame, the right-side of the bag and the CPU are always found at the same location with respect to each other across images (they are stationary), they are clustered together as the same object. Ours being an unsupervised approach, this artifact is expected, even natural, since there is in fact no evidence indicating these entities to be separate objects.

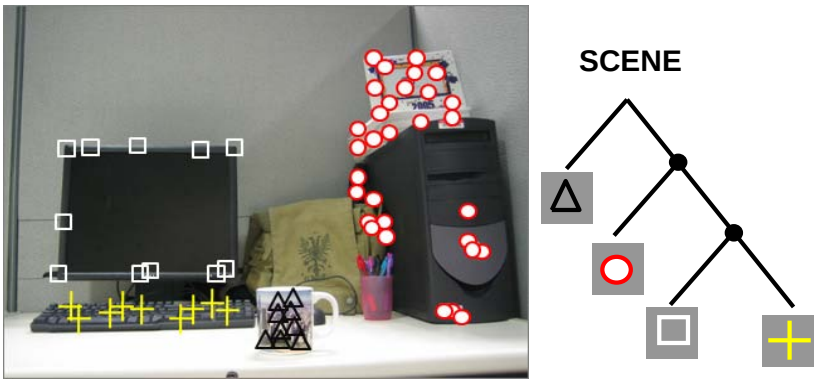


Figure 4.10: Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to an object in the foreground. Right: The corresponding learnt hSO which captures meaningful relationships between the objects.

4.5.2.2 Photo grouping

We consider an example application where the goal is to learn the semantic hierarchy among photographs. This experiment is to demonstrate the capability of the proposed algorithm to truly capture the semantic relationships, by bringing users in the loop, since semantic relationships are not a very tangible notion. We present users with 6 photos: 3 outdoor (2 beaches, 1 garden) and 3 indoor (2 with a person in an office, 1 empty office). These photos can be seen in Figure 4.11. The users were instructed to group these photos such that the ones that are similar are close by. The number of groups to be formed was not specified. Some users made two groups (indoor vs. outdoor), while some made four groups by further separating these two groups into two each. We took pictures that capture 20 such arrangements. Example images are shown in Figure 4.12. We use these images to learn the hSO. The results obtained are shown in Figure 4.13.

We can see that the hSO can capture the semantic relationships among the images, the general (indoor vs. outdoor) as well as more specific ones (beaches vs.

Figure 4.11: The six photos that users arranged.

garden) through the hierarchical structure. It should be noted that the content of the images was not utilized to compute the similarity between images - this is based purely on the user arrangement. In fact, it may be argued that although this grouping seems very intuitive to us, it may be very challenging to obtain this grouping through low level features extracted from the photos. Such an hSO on a larger number of images can hence be used to empower a content based digital image retrieval system with the users' semantic knowledge. In such a case a user-interface, similar to [109], may be provided to users and merely the position of each image can be noted to learn the underlying hSO without requiring feature extraction and image matching. In [109], although user preferences are incorporated, a hierarchial notion of interactions is not employed which provides

Figure 4.12: A subset of images of the arrangements of photos that users provided for which the corresponding hSO was learnt.

much richer information.

4.5.2.3 Quantitative results

In order to better quantify the performance of the proposed learning algorithm, a hierarchy among objects was staged i.e. the ground truth hSO is known. As shown in the example images in Figure 4.14, two candy boxes are placed mostly next to each other, a post-it-note around them, and an entry card is tossed arbitrarily. Thirty such images were captured against varying cluttered backgrounds. Note the rotation and change in view point of the objects, as well as varying lighting conditions. These were hand-labeled so that the ground truth assignments of the feature points to different nodes in the hSO are known and accuracies can be computed. The corresponding hSO was learnt from the unlabeled images. The results obtained are as seen in Figure 4.15. The feature points have

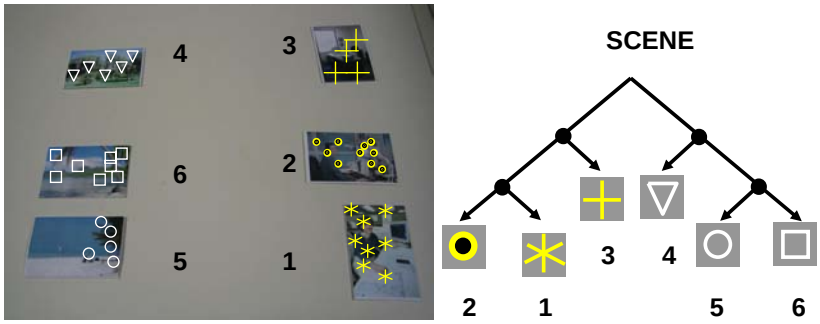


Figure 4.13: Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to a photograph. Right: The corresponding learnt hSO which captures the appropriate semantic relationships among the photos. Each cluster and photograph is tagged with a number that matches those shown in Figure 4.11 for clarity.

Figure 4.14: A subset of images of staged objects provided as input to learn the corresponding hSO.

been clustered appropriately, and the learnt hSO matches the description of the ground truth hSO above. The clutter in the background has been successfully eliminated. Quantitative results reporting the accuracy of the learnt hSO, measured as the proportion of features assigned to the correct level in the hSO, with varying number of images used for learning are shown in Figure 4.16. It can be seen that with significantly few images a meaningful hSO can be learnt. It should be noted that this accuracy simply reports the percentage of features detected as foreground that were assigned to the right levels in the accuracy. While it penalizes background features considered as foreground, it does not penalize dropping foreground features as background and hence not considering them in the hSO.

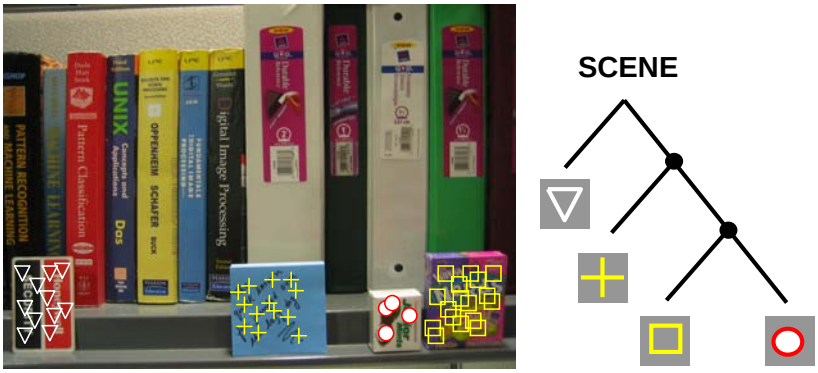


Figure 4.15: Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to an object in the foreground. Right: The corresponding learnt hSO which matches the ground truth hSO.

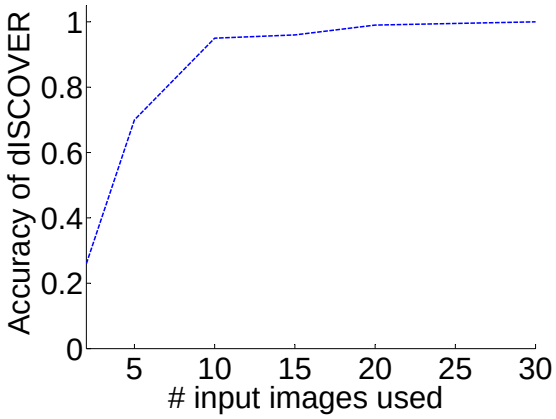


Figure 4.16: The accuracy of the learnt hSO as more input images are provided.

Visual quality of results indicate that such a metric suffices. In less textured objects the accuracy metric would need to be reconsidered.

4.6 hSO to provide context

Consider the hSO learnt for the office scene in Section 4.5.2.1 as shown in Figure 4.17. Consider an image of the same scene (not part of the learning data) as shown in Figure 4.18 which has significant occlusions (real on the keyboard,

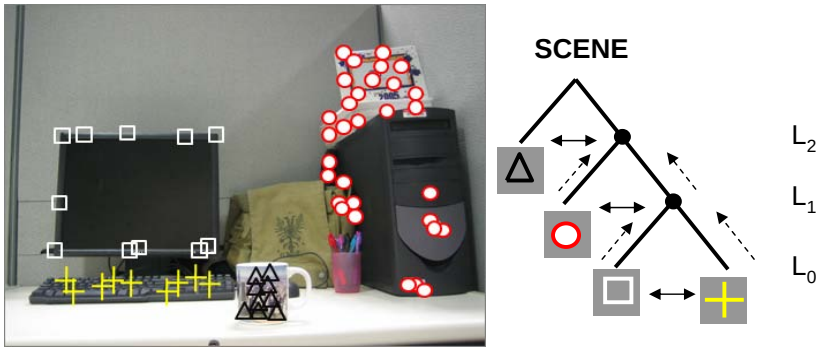


Figure 4.17: The simple information flow used within hSO for context for proof-of-concept. Solid bi-directional arrows indicate exchange of context. Dotted directional arrows indicate flow of (refined) detection information. The image on the left is shown for reference for what objects the symbols correspond to.

and synthetic on the CPU and mug). We wish to detect (we use *detection* and *localization* interchangeably) the four foreground objects.

The leaves of the hSO hold the clouds of features (along with their locations) for the corresponding objects. To detect the objects, these are matched with features in the test image through geometrically consistent correspondences similar to that in Section 4.4.2. Multiple candidate detections along with their corresponding scores are retained, as seen in Figure 4.19 (left). The location of a detection is the centroid of the matched features in the test image. The detection with the highest score is determined to be the final localization. Due to significant occlusions, background may find candidate detections with higher scores and hence the object would be incorrectly detected, as seen in Figure 4.20 (left), where three of the four objects are incorrectly localized.

In the presence of occlusion, even if a background match has a higher score, it will most likely be pruned out if we consider some contextual information (prior). To develop some intuition, we present a simple greedy algorithm to

Figure 4.18: Test image in which the four objects of interest are to be detected. Significant occlusions are present.

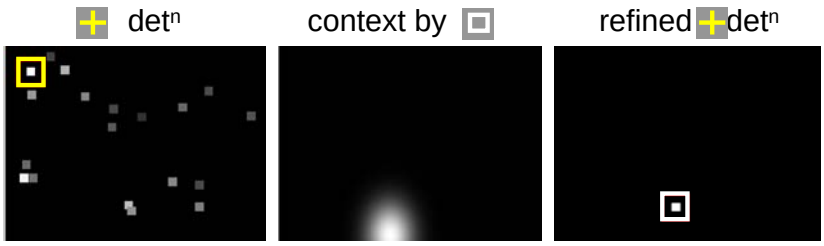


Figure 4.19: Left: candidate detections of keyboard, along with the max score (incorrect) detection. Middle: context prior provided by detected monitor. Right: detections of keyboard after applying context from monitor along with the max score (correct) detection. The centers of the candidate detections are shown.

apply hSO to provide this contextual information for object localization. The flow of information used to incorporate the context is shown in Figure 4.17. In the test image, candidate detections of the foreground objects at the lowest level (L_0) in the hSO structure are first determined. The context prior provided by each of these (two) objects is applied to the other object and these detections are pruned/refined as shown in Figure 4.19. The distribution in Figure 4.19 (middle) is strongly peaked because it indicates the relative location of the keyboard with respect to the monitor, which is quite predictable. However, the distribution

Figure 4.20: Detections of the 4 objects without context (left) - 3 of 4 are incorrect due to significant occlusions. Detections with context (right) - all 4 are correct.

of the absolute location of the keyboard across the training images as shown in Figure 4.9 is significantly less peaked. The hSO allows us to condition on the appropriate objects and obtain such peaked contextual distributions. This refined detection information is passed on to the next higher level (L_1) in the hSO, which constitutes the detection information of the super-object containing these two objects, which in turn provides context for refining the detection of the other object at L_1, and so on.

The detection results obtained by using context with this greedy algorithm is shown in Figure 4.20 (right) which correctly localizes all four objects. The objects, although significantly occluded, are easily recognizable to us. So the context is not hallucinating the objects entirely, but the detection algorithm is amplifying the available (little) evidence at hand, while enabling us to not be distracted by the false background matches.

We now describe a more formal approach for using the hSO for providing context for object localization, along with thorough experiments. We also compare the performance of hSO (tree-structure) to a fully connected structure.

4.6.1 Approach

Our model is a Conditional Random Field where the structure of the graphical model is the same as the learnt hSO. Hence, we call our graphical model an hSO-CRF. The nodes of the hSO-CRF are the nodes of the hSO (the leaves being the objects and intermediate nodes being the super-objects). The state of each node is one of the location grids in the image. Our model thus assumes that every object is present in the image exactly once. Future work involves generalizing this assumption and making use of the co-occurrence statistics of objects that can be learnt during the learning stage to aid this generalization.

Say we have N nodes (entities) in the hSO-CRF. The location of the i^th entity is indicated by l_i. Since the image is divided into a $G \times G$ grid, $l_i \in \{1, \ldots, G^2\}$. We model the conditional probability of the locations of the objects $L = (l_1, \ldots, l_{G^2})$ given the image as

$$P(\mathbf{L}|\mathbf{I}) = \frac{1}{Z} \prod_{i=1}^{N} \Psi_i(l_i) \prod_{(i,j) \in E} \Phi_{ij}(l_i, l_j), \tag{4.2}$$

where Z is the partition function, and E is the set of all edges in the hSO-CRF. The data term $\Psi_i(l_i)$ computes the probability of location of the i^{th} entity l_i across the entire image I. The pair-wise potentials $\Phi_{ij}(l_i, l_j)$ capture the contextual information between entities using the learnt relative location statistics from the learning stage.

4.6.1.1 Appearance

To compute our data term $\Psi_i(l_i)$ for the leaves of the hSO-CRF, we first match the object models stored at the leaves of the hSO to the test image as explained earlier,

to obtain a detection map as shown in Figure 4.19 (left). For each bin in the grid, we compute the maximum matching score, which is then normalized to obtain a distribution $p(l_i|I)$. Our data term (node potential) is then $\Psi_i(l_i) = p(l_i|I)$, which is a vector of length G^2. The data term for the nodes corresponding to the super-objects are set to a uniform distribution over all the bins.

4.6.1.2 Context

The edge-interactions $\Phi_{ij}(l_i, l_j)$ capture the contextual information between the i^{th} and j^{th} entities through relative location statistics. This is modeled as the empirical probability of the i^{th} and j^{th} entities occurring at locations l_i and l_j. This was learnt through MLE counts during the learning stage.

We use Loopy Belief Propagation to perform inference on the hSO-CRF using a publicly available implementation [28]. After convergence, for each object, the bin with the highest belief is inferred to be the location of object. Generally, we are not interested in the location of the super-objects, but those can be inferred similarly if required.

4.6.2 Experimental set-up

To demonstrate the use of hSO in providing context for object localization, we wish to compare the performance of hSO-CRF in providing context, to that of a fully connected CRF (which we call f-CRF) over the objects. The f-CRF is modeled similar to equation 4.2, except in this case E consists of all the edges in the fully connected graph, and N is the number of objects and not the total number of entities i.e. the f-CRF is over the objects in the images, and hence there is no concept of super-objects in an f-CRF. The node potentials and edge

Figure 4.21: Illustrations of the two types of occlusions we experiment with: (left) uniform occlusion and (right) localized occlusion. In our experiments, the amount of occlusion is varied.

potentials of the f-CRF are computed in a similar manner as the hSO-CRF. We collect test images in a similar setting as those used to learn the hSO (since the learning is unsupervised, the same images could also be used). We collect images from the office scene (example images of which are in Figure 4.9). We test only on those images that contain all the foreground objects exactly once (which form a majority of images since the foreground objects by definition occur often). We hand labeled the locations of the foreground objects in these images so that localization accuracies can be computed using these labels as ground truth.

As demonstrated in [69], the use of context is beneficial in scenarios where the appearance information is not sufficient. We simulate such a scenario with occlusions. We consider two different forms of occlusions - a uniformly distributed occlusion and a localized occlusion. The uniformly distributed occlusion is obtained by randomly (uniformly across the image) removing detected features in the image. We show illustrations of this in Figure 4.21 (left). It should be noted that we show blacked out pixels as an illustration, in reality, instead of blacking out pixels and then detecting features (which could cause several undesirable artifacts because of the nature of the SIFT detector and descriptor), we first detect

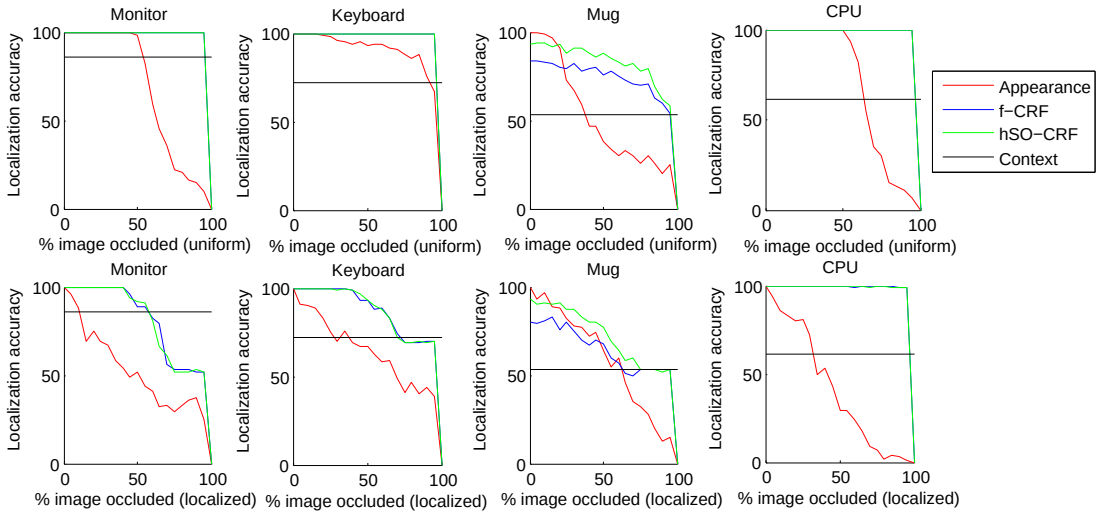


Figure 4.22: Localization results

features in the image and then randomly black out some of the features. This mimics a scenario where the images are of much lower resolution and hence fewer features are detected in the image, making the localization task hard. The second type of occlusion, is a more localized occlusion (perhaps closer to the conventional occlusions). In order to simulate this, we black out a square block of the image placed randomly in the image. An example of this is shown in Figure 4.21 (right). In both types of occlusions, we vary the amounts of occlusions added to the test images.

The results obtained are shown in Figure 4.22. We show the localization accuracies for all four foreground objects: monitor, keyboard, mug and CPU for the office scenario for which the hSO was learnt as shown in Section 4.5.2.1, for the two types of occlusions and for varying amounts of occlusions. We compare the accuracies of hSO-CRF to that of f-CRF. Recall, that the learnt hSO as shown in Figure 4.10 indicates that the monitor and keyboard are most related, followed by the CPU, and the mug was the most unrelated/unpredictable in the scene. For

more insight in the test scenario we also report accuracies of using appearance information alone (edge potentials on the hSO-CRF were set to uniform) and using contextual information alone (node potentials in the hSO-CRF for all the objects were set to uniform). The accuracies of the hSO-CRF and f-CRF are similar for most objects. And since f-CRF is a fully connected network and hence much more complex to run inference on as opposed to hSO-CRF which has a tree structure, the advantage of hSO-CRF is clear. Moreover, the accuracy of hSO-CRF for the mug is much higher than that for f-CRF. This validates our claim that f-CRF is prone to over-fitting because it explicitly models relationships among objects that may be unrelated, while the hSO-CRF models relationships only among entities that are related.

We find that in the presence of very little occlusion, appearance information alone has higher localization accuracy for the mug than both f-CRF and hSO-CRF (however, hSO-CRF has significantly higher accuracy than f-CRF). This is again because the location of a mug is unpredictable, and hence if available, the appearance information is most reliable. In general we find that the localization accuracies for the uniform occlusion are higher than for the localized occlusions. This makes intuitive sense. Also, similar to the findings of the previous chapter, we find that context provides a boost in performance only when the appearance information is weak, and not otherwise. Another observation is that the monitor and keyboard localization accuracies using both hSO-CRF and f-CRF with significant amount of localized occlusions are lower than using context alone (no appearance information). This indicates that extremely poor appearance information can hurt the performance as compared to using no appearance information at all and relying only on learnt contextual statistics. This indicates

that depending on the scenario (amount of occlusion), roles of appearance and contextual information vary. Overall, the performance of hSO-CRF is the most reliable.

4.7 Conclusion

We introduced hSOs: Hierarchical Semantics of Objects that capture potentially semantic relationships among objects in a scene as observed by their relative positions in a collection of images. The underlying entity is a patch, however the hSO goes beyond patches and represents the scene at various levels of abstractness - ranging from patches on individual objects to objects and groups of objects in a scene. An unsupervised hSO learning algorithm has been proposed. Given a collection of images of a scene, the algorithm can identify the foreground parts of the images, group the parts to form clusters corresponding to the foreground objects, learn the appearance models of these objects as well as relative locations of semantically related objects and use these to provide context for robust object detection even with significant occlusions - all automatically and entirely unsupervised. This, we believe, takes us a step closer to true image understanding. We demonstrate the need for popularity as well as geometric consistency based cues for successful extraction of multiple foreground objects. We also demonstrate the effectiveness of a meaningful hierarchical structure to provide context for object localization as compared to a fully connected network that is prone to over-fitting.

We now present our approach to learning hierarchical spatial patterns in more general settings of object categories, scene categories, complex real world scenes,

etc.

Chapter 5

How: Unsupervised Learning of Hierarchical Spatial Structures In Images

Summary

The visual world demonstrates organized spatial patterns, among objects or regions in a scene, object-parts in an object, and low-level features in object-parts. These classes of spatial structures are inherently hierarchical in nature. Although seemingly quite different these spatial patterns are simply manifestations of different levels in a hierarchy. In this work, we present a unified approach to unsupervised learning of hierarchical spatial structures from a collection of images. Ours is a hierarchical rule-based model capturing spatial patterns, where each rule is represented by a star-graph. We propose an unsupervised EM-style algorithm to learn our model from a collection of images. We show that the inference problem

of determining the set of learnt rules instantiated in an image is equivalent to finding the minimum-cost Steiner tree in a directed acyclic graph. We evaluate our approach on a diverse set of data sets of object categories, natural outdoor scenes and images from complex street scenes with multiple objects.

5.1 Introduction

Our visual world is far from random, and demonstrates highly predictable spatial patterns. These patterns may be among high-level entities such as objects in a scene (keyboards are usually below monitors), regions in a scene (sky is usually above grass), parts within an object (the engine is usually in between the two wheels of a motorcycle), or among low-level features within object-parts. These classes of spatial structures are inherently hierarchical in nature, as shown in Figure 5.1.

Previous work has used each of these levels for various tasks. For instance, patterns among object parts are used to form compositional models to aid in object recognition [3, 4, 110, 111, 112]. The relationship of objects are used to capture semantic contextual information for robust object detection/localization or image labeling [16, 9, 8, 7, 69]. Clusters of low-level features have been shown to be more discriminative than single features for object recognition [56, 113].

Although seemingly quite different, these various forms of spatial patterns can simply be viewed as manifestations of different levels in a hierarchy [114, 115, 11, 20, 86, 17, 93, 116]. It is clear that extracting this hierarchy of spatial structures could provide rich information to facilitate several vision tasks such as image classification, localization, object recognition, and others. However,

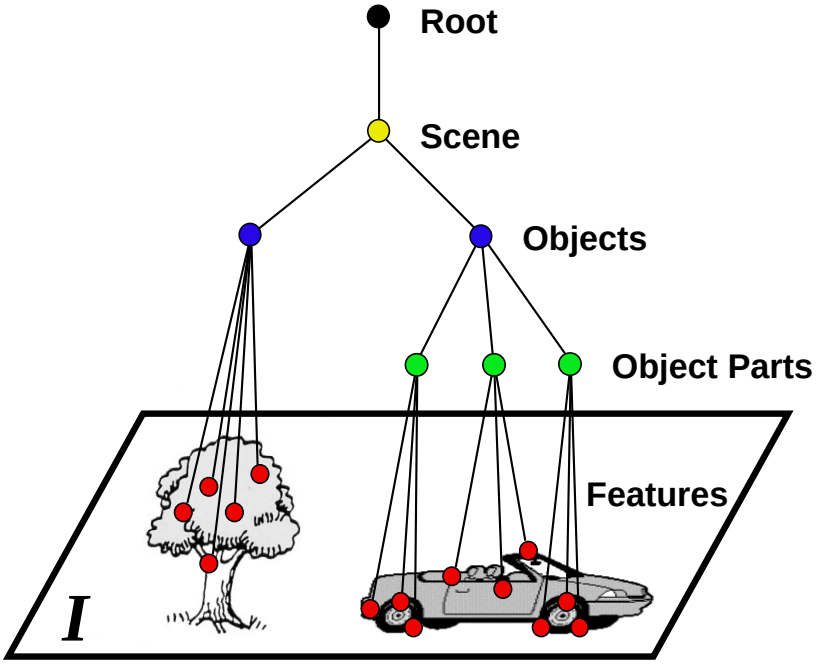


Figure 5.1: An illustration of the hierarchical spatial patterns present in an image.

learning such a hierarchy would be prohibitive if it required extensive supervision and laborious labeling of images. In this chapter, we propose a unified approach to unsupervised learning of hierarchical spatial structures [20] from a generic collection of images. We describe each spatial pattern in the hierarchy as a rule. Each rule is represented by a star-graph [4], where a child of the star-graph may be a low-level feature or another star-graph (i.e. rule), thus forming a hierarchy.

The inference problem is to determine the subset (hierarchy) of learnt rules that best explains the observed features in a given image. We impose that the set of rules that can be used to explain the image forms a tree. That is, each feature or rule can only be explained by a single parent rule. We show that determining the optimal tree that maximizes the likelihood of the image is equivalent to finding the minimum cost Steiner tree [117] in a directed acyclic graph (DAG). This

being an NP hard problem, we use an approximation algorithm proposed by Charikar *et al.* [118]. It should be noted that the structure of the optimal tree (as well as the underlying DAG) may be different for different images, and is determined automatically during inference. For computational feasibility, we reduce the number of rules considered for inclusion in the tree using a voting scheme.

The learning task is to infer a set of rules from a given collection of images in an unsupervised manner. The number of rules, the structure and parameters of each rule, and number of children of each rule are learnt automatically. Ours is an EM-style algorithm where we initialize our model (a set of rules), infer instances of them across an image collection, and update the rule parameters.

We evaluate our approach on a diverse collection of datasets ranging from a subset of the Caltech101 object categories [5], outdoor natural scene categories [1], as well as complex street scenes from the LabelMe dataset [37]. We present qualitative results through visualizations of the rules learnt and the hierarchies inferred in images. To demonstrate the behavior of the rules in the hierarchy, we quantify at each level the localization and categorization abilities of the rules. We find that higher level rules are often specific to object categories, while lower-level rules can be shared between categories. To demonstrate the utility of the learnt spatial hierarchies, we perform unsupervised clustering of the images into object categories. We report comparable accuracies to the state-of-the-art techniques.

We discuss related work next in Section 5.2. Sections 5.3, 5.4 and 5.5 describe our model, our method for inference given an image and a learnt model, and our unsupervised approach to learning the proposed model. Section 5.6 describes our experiments and presents results. Section 5.7 raises some points of discussion and

future work, followed by a conclusion in Section 5.8.

5.2 Related Work

Modeling image hierarchies and spatial structures has a long history in computer vision [119, 114, 115]. The works vary both in their representations used to encode the spatial information and their approaches for learning. We discuss both the *representations* and *learning* algorithms in turn.

Representation: Different representations based on global histograms [56], graphs [4, 110, 3] and hierarchies [114, 115, 11, 20, 17, 93, 116] have been explored in previous works. The bag-of-words model [56] uses a histogram representation which is efficient to compute and match. Graph-based methods have been proposed for recognizing individual objects using Constellation models [3] and star-graphs [110] where pair-wise spatial location statistics are captured. Graphs have also been used for context modeling in street scenes by Hoiem *et al.* [8] and segment labeling [7, 69]. Numerous hierarchical methods have been proposed. Several approaches use a fixed number of levels such as Kumar and Hebert [16] that use a two level hierarchy to model context in classification. Sudderth *et al.* [86] use hierarchies for part sharing and modeling scenes, while Murphy *et al.* [11] model the spatial relationship of objects in scenes. Other models use hierarchies of arbitrary depth. These methods can be used to model individual objects, e.g. the segment tree approach of Todorovic and Ahuja [120, 121], the method of Zhu *et al.* [122] for deformable objects and the object part discovery approach of Fidler *et al.* [123]. Other approaches attempt to model relationships between object parts within a hierarchy, such as the stochastic grammar approach

of Zhu *et al.* [116] using And-Or graphs. Finally, some approaches attempt to create hierarchies of object categories based on object appearances [124].

Learning: The level of supervision varies among the various approaches proposed in the literature. Supervised techniques [110, 116] require objects to be labeled in the images for learning. A less restrictive class of techniques called weakly-supervised [3, 111] only requires the knowledge of whether an object is present in the image or not. Several of the existing hierarchical representations are learnt in a supervised [121, 125, 126] or semi-supervised way [122, 127], or learn only part of the model from training data. For instance a structure of the hierarchy may be given and only the parameters are learnt from data [128], or the entire model is given and the task is to only infer the model in images [129]. Finally, unsupervised techniques require only a set of unlabelled images for learning. Unsupervised techniques have been proposed for bag-of-words models [56] and models that learn spatial structure [130, 112, 131].

5.3 Model

Our model is a hierarchy of rules. Each rule describes a spatial pattern, and is represented as a star-graph. Just as in language modeling, a sentence is modeled as a parse tree, we consider an image to have an associated tree formed by the subset of rules that best explains the observed features in the image. The leaves of the tree are the observed features, and the intermediate nodes are the higher-order spatial-patterns (instantiations of the rules), which we call image-parts. An image-part could correspond to higher order features, object-parts, objects, groups of objects or a scene. The inference task is to find the set of image-parts

that best explain the features in an image, given a set of rules. We first introduce some notation.

Each feature $f \in F$ is an instantiation of a codeword at a certain location, denoted as a pair (c_f, l_f), where $c_f \in C$ and l_f is the location of feature f. C is the dictionary or vocabulary of all possible discrete appearances of the low-level features (codewords). Each rule r, as shown in Figure 5.4, is defined by a certain structure and associated parameters denoted by θ_r. A rule defines a star graph with associated children $Ch(r)$. A child $x \in Ch(r)$ may be either a codeword c, or a another rule r, i.e. $x \in C \cup R$. Allowing rules to be children of rules enables the formation of hierarchies. Not all children in a rule may be instantiated in an image. The parent of x is denoted as $Pa(x)$. The rule parameters θ_r contain both the occurrence probability for a child $\Pr(x|r)$ and the location probability $\Pr(l_x|r)$, where l_x is the location of the child relative to the parent. We model the location probability using a Gaussian with an associated mean and covariance.

Finally, we define a *background* or *prior* image-level rule, indicated by r_0, whose definition encompasses all codewords and rules i.e. $Ch(r_0) = C \cup R$. The parameters for this rule are the prior probabilities (for instance, the marginal probability of observing a certain codeword or rule at a certain location in an image). From here on, we include r_0 in the set of all rules R. r_0 acts as the root node, similar to the node corresponding to a sentence in language modeling.

We define the set of instantiated image-parts as H. A tree $T = \{V, E\}$ for image I consists of a set of vertices V and edges E. The vertices are the union of the image-parts and features, i.e. $V = H \cup F$. The edges E indicate the set of children $Ch(v)$ for each vertex $v \in V$. If v corresponds to a feature then $Ch(v) = \emptyset$. If v corresponds to a rule r_v, the rule's children $Ch(r_v)$ may or may

not be instantiated. A child $x \in Ch(r_v)$ is instantiated if $x \in Ch(v)$, i.e. x is instantiated if there exists a vertex $v' \in Ch(v)$ corresponding to x. The parent of a vertex v is defined as $Pa(v) \in H$, and its location in the image by l_v.

With this notation, we can now introduce our model. Given an image with a set of observed feature F, our goal is to find a tree T such that each feature f corresponds to a leaf in the tree. Each feature may only be explained once, i.e. it may only have one parent, and each feature must be directly or indirectly attached to the root node corresponding to rule r_0. The intermediate nodes in the tree are image-parts corresponding to instantiated rules r_v. An image I may have numerous feasible trees, and the likelihood of the image under any such tree T is given by:

$$\Pr(I|T,R) = \prod_{v}^{V} \prod_{x}^{Ch(r_v)} \rho(x,v) \tag{5.1}$$

where the value of $\rho(x,v)$ depends on whether the child x of r_v is instantiated in the tree T.

$$\rho(x,v) \;=\; \left\{ \begin{array}{ll} \Pr(x|r_v)\Pr(l_x|r_v) & x \in Ch(v) \\ 1 - \Pr(x|r_v) & \text{otherwise} \end{array} \right\} \tag{5.2}$$

Before we present our approach to unsupervised learning of our model R from a collection of images, we describe our approach to the inference problem.

5.4 Inference

The inference problem entails determining the tree T^* that best explains the observed set of feature F in a particular image I, given our learnt model R. This can be formulated as

$$T^* = \text{argmax} \Pr(I|T, R) \tag{5.3}$$

As stated earlier, a tree is formed of image-parts (hidden) as intermediate nodes and features (observed) as leaves. The task is to determine which and at what location rules from our model should be instantiated in the image, such that the observed features are best explained. Considering a dense sampling of potential locations for every rule in the model would result in a very large number of potential image-parts to be considered, making this task computationally infeasible. Instead, we select a sparse set of likely locations for each rule. While this greatly increases the computational efficiency, the optimally of the tree cannot be guaranteed.

We first present our approach for determining the subset of optimal image-parts from a pool of potential image-parts such that the resulting tree best explains the image. This is followed by a section describing how the initial set of potential image-parts is found.

5.4.1 Inferring the tree

Having computed a set of potential image parts $\tilde{H}$, we need to determine the subset of parts $H \subset \tilde{H}$ that best explains the image in the form of a tree. An image is considered to be explained if all the observed features in the image are assigned to some image-part. All image-parts that are retained must be directly or indirected connected to the root node corresponding to r_0.

The set of all possible assignments of features to image-parts and image-parts

to image-parts forms a weighted directed acyclic graph (DAG) G. Our goal is to find a tree $T \subset G$ such that Equation (5.1) is maximized. To achieve this goal we map our problem to that of a Steiner tree [117]. A minimum cost Steiner tree is the same as a Minimum Spanning Tree (MST) except some vertices in the graph do not need to be in the final tree. For our task, all image-parts not corresponding to the root node are considered optional. To map our problem to a the Steiner tree, we need to define a set of edge weights for every edge in G. Since Equation (5.1) is dependent on uninstantiated parts that may not exist in T, it cannot be directly applied for computing edge weights. Instead we perform the following manipulations on equation (5.1) to find our set of edge weights. First, we define two helper functions $\alpha(x, v) = \Pr(x|r_v)\Pr(l_x|r_v)$ and $\beta(x, v) = 1 - \Pr(x|r_v)$ corresponding to the two parts of Equation (5.2). If x_v corresponds to the rule or codeword at vertex v, we find:

$$\begin{aligned}\Pr(I|T,R) &= \left(\prod_{v}^{V}\prod_{v'}^{Ch(v)}\alpha(x_{v'},v)\right)\times \\ &\left(\prod_{v}^{V}\prod_{x}^{Ch(r_v)\setminus Ch(v)}\beta(x,v)\right) \qquad (5.4)\\ &= \left(\prod_{v}^{V}\prod_{v'}^{Ch(v)}\frac{\alpha(x_{v'},v)}{\beta(x_{v'},v)}\right)\times \\ &\left(\prod_{v}^{V}\prod_{x}^{Ch(r_v)}\beta(x,v)\right) \qquad (5.5)\end{aligned}$$

Since the value of $\prod_{x}^{Ch(r_0)}\beta(x, v_0)$ for the root node is constant for all trees, we can rewrite the second part of Equation (5.4) as:

$$\prod_{v}^{V}\prod_{x}^{Ch(r_v)}\beta(x,v) \propto \prod_{v}^{V}\prod_{v'}^{Ch(v)}\prod_{x'}^{Ch(r_{v'})}\beta(x',v') \tag{5.6}$$

As a result:

$$\Pr(I|T,R) \propto \prod_{v\in V}\prod_{v'}^{Ch(v)}\left(\frac{\alpha(x_{v'},v)}{\beta(x_{v'},v)}\prod_{x'}^{Ch(v')}\beta(x',v')\right) \tag{5.7}$$

We then assign our edge weights $\omega(v',v)$ for all $v,v' \in V$ such that $v = Pa(v')$ as:

$$\omega(v',v) = -\log\left(\frac{\alpha(x_{v'},v)}{\beta(x_{v'},v)}\prod_{x'}^{Ch(v')}\beta(x',v')\right) \tag{5.8}$$

Using Equation (5.8) we can assign edge weights to every edge in G and solve for the minimum cost Steiner tree. That is, the tree with minimum edge weights that connects each feature to the root node, using any subset of image-parts. Since this has been shown to be a NP-hard problem, we use the approximation algorithm proposed by Charikar *et al.* [118]. For a graph G, the minimum cost Steiner tree is the optimal solution for Equation (5.3) except in special cases when multiple instantiations of a rule's child are found. In these cases, we simply choose the most likely instantiation of the child and add the rest to the root node.

5.4.2 Determining candidate locations

In the previous section we discussed how to find the optimal tree given a candidate set of image-part locations. In this section we describe how the candidate set is found. The candidate locations of rules are determined through a voting mechanism. A map is thus created over the entire image, indicating the likelihood of the rule occurring at that location. The peaks in this distribution are then computed using non-local-maxima-suppression, which form candidate part locations. These distributions for the rules are computed in order of their associated levels, where the lowest level parts (codewords) vote for the first level parts, which in turn vote for the second level parts, and so on. The level of a rule is recursively defined as one more than the maximum level of all its children. The level of codewords is arbitrarily defined to be 0.

The cumulative votes $\xi(v)$ of all children of potential vertex v are computed as:

$$\xi(v) = \sum_{x}^{Ch(r_v)} \alpha(x, v) \tag{5.9}$$

This additive form allows for our framework to be robust to missing children and occlusions. This provides an advantage over other methods such as pictorial structures [4] that are not robust to occlusions. By using a subset of image-part locations, the globally optimal tree for an image may not be found. However, it allows for the computational feasibility of the algorithm.

5.5 Learning

We use an EM-style approach for unsupervised learning of rules for image parts. A set of rules is first initialized. Then we iteratively infer the rules in our image data set using the Steiner tree formulation described above, update the rule parameters given their found instantiations and repeat. In addition, we add and remove rules during each iteration. Example rules are illustrated for the face and motorbikes data sets in Figures 5.4 and 5.3

We initialize each rule by randomly selecting an image and location. Children are assigned to the rule based on the codewords that exist in a certain spatial neighborhood. In all our experiments, we randomly selected 10 codewords in a spatial neighborhood equal to a quarter of the image size. The mean relative location of the children is set according to their location in the image, the covariance matrix is set to a diagonal matrix with entries equal to one third the image size and the probability of occurrence is set to 0.25. This gives us our initial model R. Initially all rules belong to the first level. As the learning proceeds, higher level rules are added in a similar manner.

Given a set of rules, a new set of instantiated image-parts are inferred, and the rule parameters are updated. First, every non-root vertex v in the inferred tree T is assigned to an image-part. If a vertex was assigned to the root node by the Steiner tree, then it is reassigned to the nearest image-part of higher ranking if one exists. Next, the vertices corresponding to the same rules or features are clustered using meanshift. Each cluster is then assigned as a child to the parent rule with the appropriate occurrence probability, mean and covariance. Clusters with fewer than ten members are removed. It is worth noting that multiple

instances of the same codewords or rules can be added to a parent rule. This allows a rule to have multiple children with similar appearances, such as the two wheels of a motorbike.

Rules may also be removed and added during each iteration. If a rule was not inferred in at least 10 images, it is removed from the set of rules. New rules are added in a similar manner as in initialization. However, image-parts are now inferred, so a hierarchy of parts may form. It is possible to also limit rules to only have image-parts as children. This explicitly encourages higher level rules to be formed.

In all our experiments, we used 30 iterations for each level, and computed a total of two levels. Rules were added only once every three iterations to allow the existing rules to stabilize before new rules were added. The number of added rules varied from 4 to 18 depending on the database size.

5.6 Experiments and Results

We present results of our unsupervised learning algorithm on a variety of datasets containing object categories, natural outdoor scene categories as well as complex street scenes with multiple objects. We present qualitative results through visualizations of the learnt rules. We explore the behavior of rules at different levels in the hierarchy for categorization and localization.

For categorization, we compare the bag-of-words descriptor based on codewords to one based on the inferred rules. While a bag-of-rules descriptor captures which rules were instantiated in an image, it does not capture which children of the rule were instantiated to support the rule. We use the parse tree inferred for

the image as a descriptor to capture this information. The length of the descriptor is the same as the number of parent-child relationships in the learnt rules, where an element is set to 1 if the corresponding parent child relationship was instantiated in the image.

For unsupervised clustering of images into object categories, we use PLSA [56], k-means and normalized cuts [107] on a fully connected graph, where each node corresponds to an image, and a node is connected to its five nearest neighbors computed using normalized dot-product of the image descriptors. For supervised classification of images, we use a linear SVM.

5.6.1 Faces vs Motorbikes: SIFT

For illustration and intuition-building purposes, we first present results on a dataset composed of 100 random images each from the Face and Motorbike categories of the Caltech101 data set [5]. We use the SIFT [38] descriptor on interest points as our low-level descriptor, along with a dictionary of 200 visual words. Our learning procedure learnt 15 first level rules, and 2 second-level rules.

An illustration of the rules learnt can be seen in Figure 5.2. We see that the second level rules correspond to the object category, while their children (first level rules) correspond to object parts (chin, cheek, wheel, etc.), as also seen in Figure 5.3. Some of these parts are shared across both categories, while some are specific to each category.

As seen in Figure 5.2, we see that at higher levels, the objects are better localized. A similar trend is seen for categorization as seen in Figure 5.5. To quantify this behavior, we use the occurrence of each part individually to categorize the

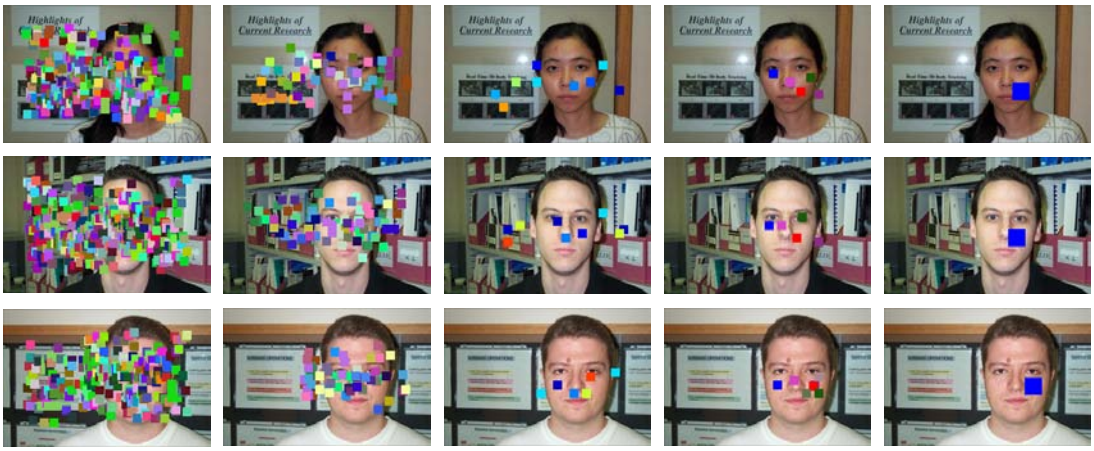


Figure 5.2: The first column illustrates all the visual words observed in the image. The second column depicts the subset of codewords that were assigned to a higher level part. The third column depicts the location of the first level parts, a subset of which (fourth column) support a second level part which are shown in the last column.

image as well as localize the foreground. For the purpose of evaluation, we considered faces to be the positive class for categorization; and hand-labeled bounding boxes around faces for localization (and the rest of the image as the negative class). For localization, we find that the sensitivity of the different levels of parts shown in Figure 5.2 is 0.44, 0.56, 0.61, 0.69 and 0.94, while the specificity is 0.61, 0.76, 0.82, 0.95 and 0.99. Similar trends were found for categorization. The higher level spatial patterns provide more accurate categorization and localization. It should be noted that we only penalize the firing of a part on background, and not the assignment of a foreground codeword to background.

Using the bag-of-words model followed by k-means clustering gives us categorization accuracies of 93.5%. Our bag-of-rules descriptor can classify each image correctly. SIFT features alone can separate faces from motorbikes accurately, and hence the advantage of using higher order spatial patterns is not clear. We ex-

Figure 5.3: Patches extracted around instantiation of three first level rules for the faces and motorbikes data set. The first rule is specific to faces, the second one is specific to motorbikes, while the third one is shared across categories.

Figure 5.4: Example rules learnt by our algorithm from an unlabeled collection of face and motorbike images. The first column illustrates the structure of these first level rules and the relative spatial locations of its children. The last four columns show instantiations of the rules in example images.

periment with edge features (at 4 orientations and 6 scales, forming a dictionary of 24 codewords) as shown in Figure 5.6 and find that using our learnt rules the categorization accuracy increases from 55% using bag-of-words to 81.7%.

5.6.2 Six object categories

Unsupervised clustering of images into object categories is one potential application of the proposed model. To this end, we evaluate our approach on 100

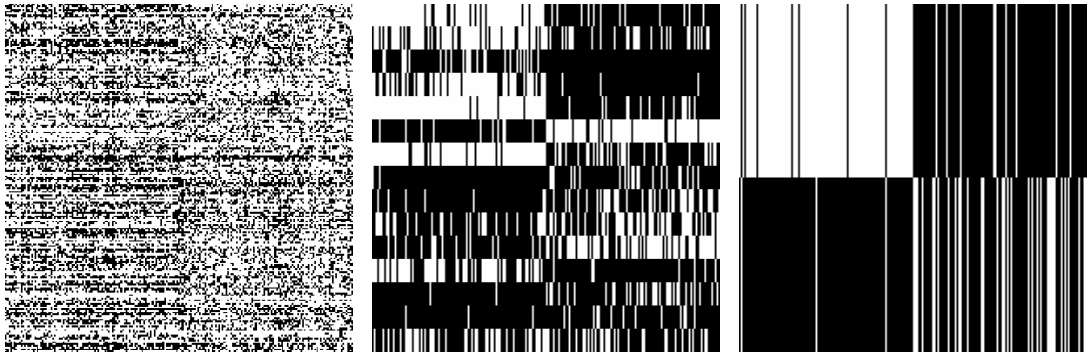


Figure 5.5: On the left is the occurrence matrix of the codewords (rows) in the face (left half of the matrix) and motorbike images (right half of the matrix). It is evident that codewords are not specific to either category. The middle plot is the occurrence matrix of the first level rules, where the distinction between the two categories improves, followed by the occurrence matrix of the second level rule.

Table 5.1: Categorization accuracy (%) using 100/30 images per category

	Kmeans	PLSA	NNgraph	SVM
Words	70.7/72.8	80.5/78.3	86.5/84.7	93.3/91.8
Rules	85.2/86.5	84.7/85.6	94.2/92.6	91.3/90.7
Both	73.9/74.3	82.6/84.7	90.1/88.8	95.8/93.2
Tree	88.1/89.5	85.1/88.2	95.0/93.5	91.3/89.8

random images from 6 object categories (faces, motorbikes, airplanes, car-rear, watches and ketches) from Caltech101 [5], similar to the recent work of Kim *et al.* [131]. Our accuracies are reported in Table 5.1, and are comparable. A total of 61 first level rules, and 12 second level rules were learnt. On average, the first level rules had 9 children, and the second level rules had 3. We also train our model using only 30 images per category, and obtain comparable accuracies.

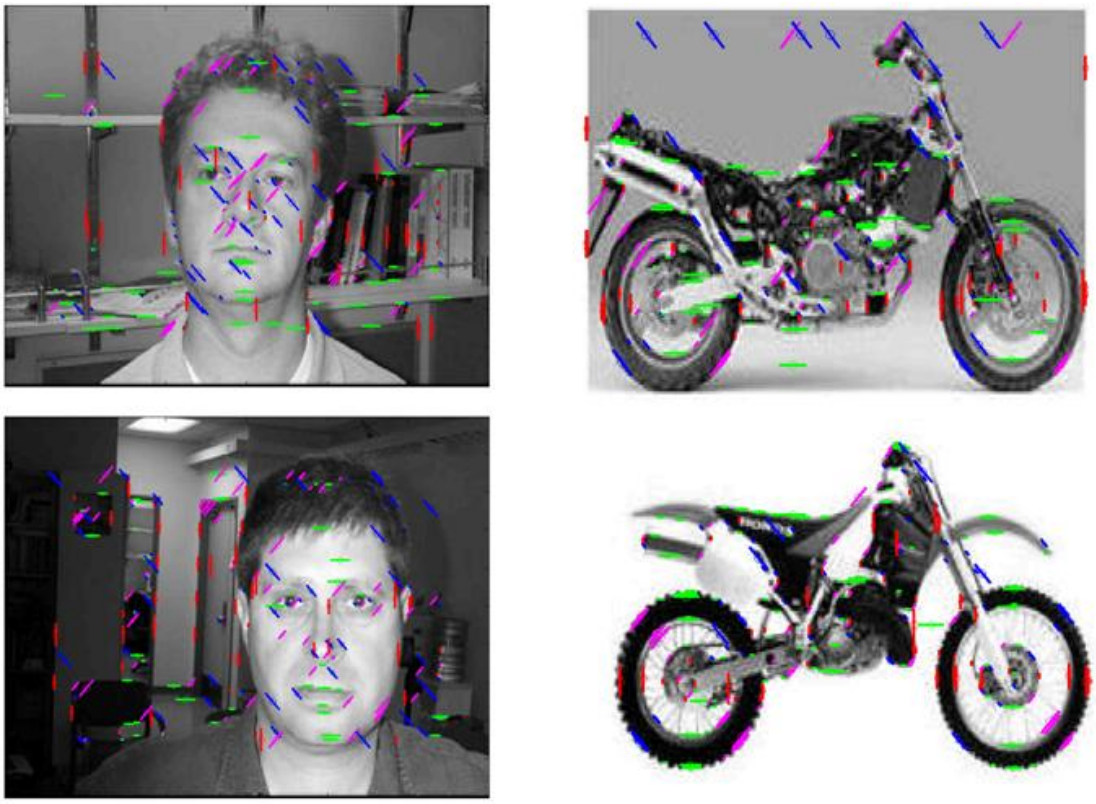


Figure 5.6: The edge features used as low level features to learn our higher-order parts. Only the edge information displayed in color was fed to the algorithm, discarding the rest of the image

5.6.3 Scene categories

We experiment with a dataset containing 150 images from the outdoor scene recognition dataset by Torralba *et al.*. [1]. We segmented these images to obtain on average 10 segments per image using the segmentation algorithm of Felzenszwalb *et al.* [33]. Each segment was described with its average RBG color vector. These color descriptors from all the segments from all images were clustered to form a dictionary of 25 codewords. Using our learning algorithm on these images, we were able to find only first level rules, whose spatial extent was often the entire image. This is intuitive behavior for this dataset, where a deeper

hierarchy is non-existent. A total of 17 rules were learnt. A visualization of a subset of rules learnt can be seen in Figure 5.7. We can see that images with consistent spatial layout of colors are grouped together. In the last two rows (first and last image respectively), we see that the color histogram of the images may be similar, however the spatial layout of the colors distinguish them from each other. This demonstrates the ability of the algorithm to find meaningful patterns in a seemingly unstructured collection of images, which could have several interesting applications in visualization, finding canonical images, finding similar images that can share scribbles for the task of co-segmentation, etc.

5.6.4 Street scenes

We select 66 images from the street scenes in the LabelMe dataset [37]. We use SIFT features with a dictionary of 200 codewords. 25 first level and 8 second rules were learnt. Illustrations of these rules are shown in Figures 5.8 and 5.9. It can be seen that the features supporting the first level rules are consistently found on objects/regions of the image, and the second level rules correspond to objects (cars, trees, buildings), or combine contextually meaningful objects (cars and buildings). Further visualizations are shown in Figure 5.10 , where desirable outcomes such as instantiation of the same rule on repeating window patterns on the buildings is seen. We also compare our approach to PLSA on this dataset in Figure 5.11. While it is hard to find a pattern in these results, our learnt rules correspond to distinct objects such as buildings, cars and trees.

To quantify the quality of our learnt parts, we consider three categories from our street scene images: cars, trees, and buildings. We label each instantiation

of these categories in our images. We train a binary logistic regression classifier on our rules for each of these three categories. We pick the rule with the highest weight in the classifier, and treat it as a textitdetector for that category. We compute the detection accuracy of the object i.e. the percentage of the the object instantiations that contained an instantiation of the part. These are shown in Figures 5.12. We also compute the precision of the part i.e. the percentage of the part instantiations that occurred on the object category. These are shown in Figures 5.13. As can be seen, even though the parts are learnt in an unsupervised manner from a collection of images, they are very well correlated with semantically meaningful object categories in the dataset.

To evaluate the specificity of the learnt rules to the data it is learnt on as opposed to noise, we infer the the rules learnt on the street scene images on 66 background images (from the Catech101 dataset). Figure 5.14 depicts the histogram of the number of rules instantiated in the background images, as compared to the "foreground" street scene images. We can see that the histograms are well separated, and simply by counting the number of rules instantiated in an image, it can be separated into the street scene vs. background.

5.7 Discussion and Future Work

This work describes a hierarchical representation of the image that inherently allows for the sharing of low-level image parts. Occlusions are also explicitly modeled. However, since we represent our rules using star-graphs we assume the children of every vertex in our tree are independent. This does not allow us to capture higher order relationships among parts.

The efficiency of our approach is mainly limited by the choice of algorithm for solving the Steiner tree. Quasi-polynomial approximate algorithms have been developed to solve the general Steiner tree problem [118] that is known to be NP-Hard. Unfortunately, these algorithms are still too inefficient for large graph sizes. Given the specialized structure of our problem, it may be possible to create better approximate algorithms.

The accuracy of our approach is limited by the choice of low-level features. Features such as SIFT [38] already contain significant structural information. More primitive features such as edges may provide increased robustness to background clutter and shape ambiguity. These primitive features may also require more levels in the hierarchy to find coherent objects.

5.8 Conclusion

In this chapter, we proposed an unsupervised method for learning hierarchical spatial structures in images. Our model consists of a set of rules modeled as star graphs, in which the children of each rule may be another rule or a low-level feature. The structure and parameters of the rules are learnt automatically. Given an image, a set of rules is inferred that best predicts the occurrence of the low-level features in the image. This subset of rules form a tree, and inference is accomplished by mapping the problem to that of finding the minimum cost Steiner tree in a directed acyclic graph, for which approximate algorithms exist.

We provide several results on various data sets including six Caltech 101 object categories, an outdoor scene data set, and a real-world street scene image collection from the LabelMe data set. Quantitative and qualitative results are

provided. The unsupervised approach is shown to discover categories in images containing just one object, as well as multiple objects.

Figure 5.7: Each row corresponds to a rule learnt from an unstructured collection of outdoor scene category images. For each rule we show 7 random images that instantiated this rule. It can be seen that the images are consistent in the spatial distribution of their colors.

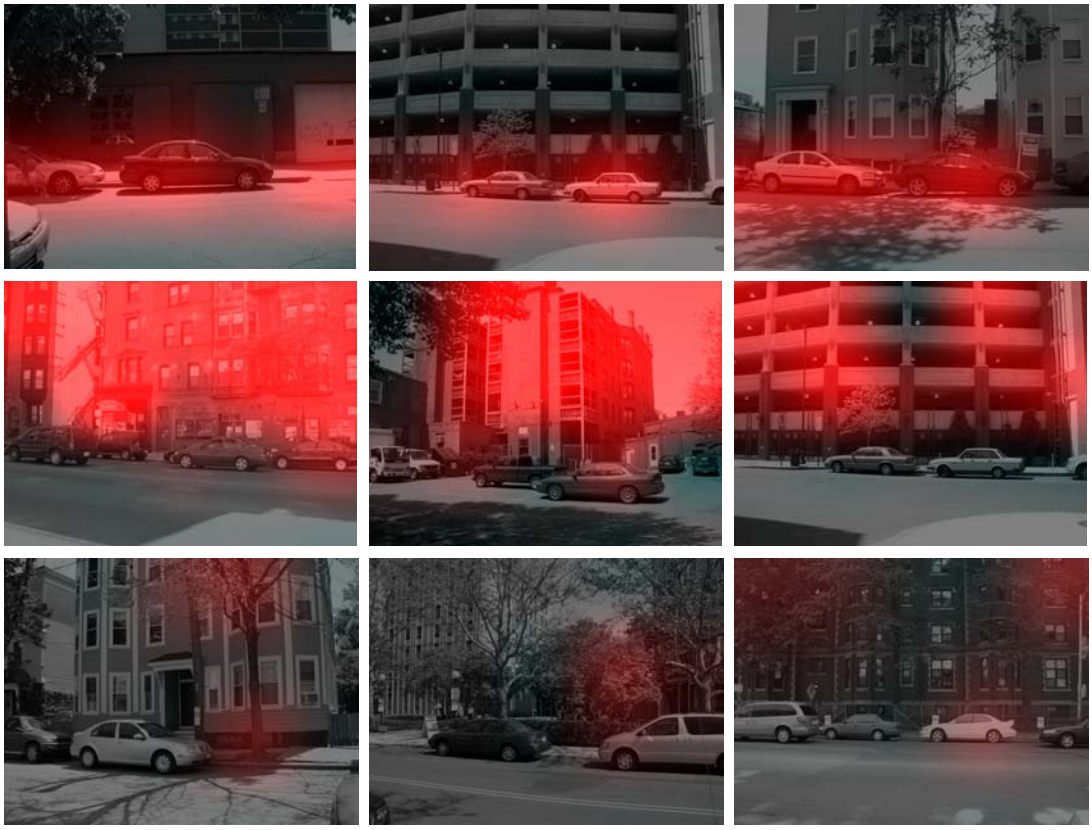


Figure 5.8: An illustration of three first level rules (rows) learnt from street scene images. We highlight the regions of the image with a high density of features that support each rule. In general, the first rule corresponds to buildings, the second one to cars and the third one to trees.

Figure 5.9: An illustration of four second level rules learnt from street scene images. The first level rules that support the second level rule are shown. The first rule (row) corresponds to cars (note the instantiation of the same rule twice for the two cars in the last column), the second rule corresponds to trees, the third to buildings and the fourth combines the cars and buildings in one rule.

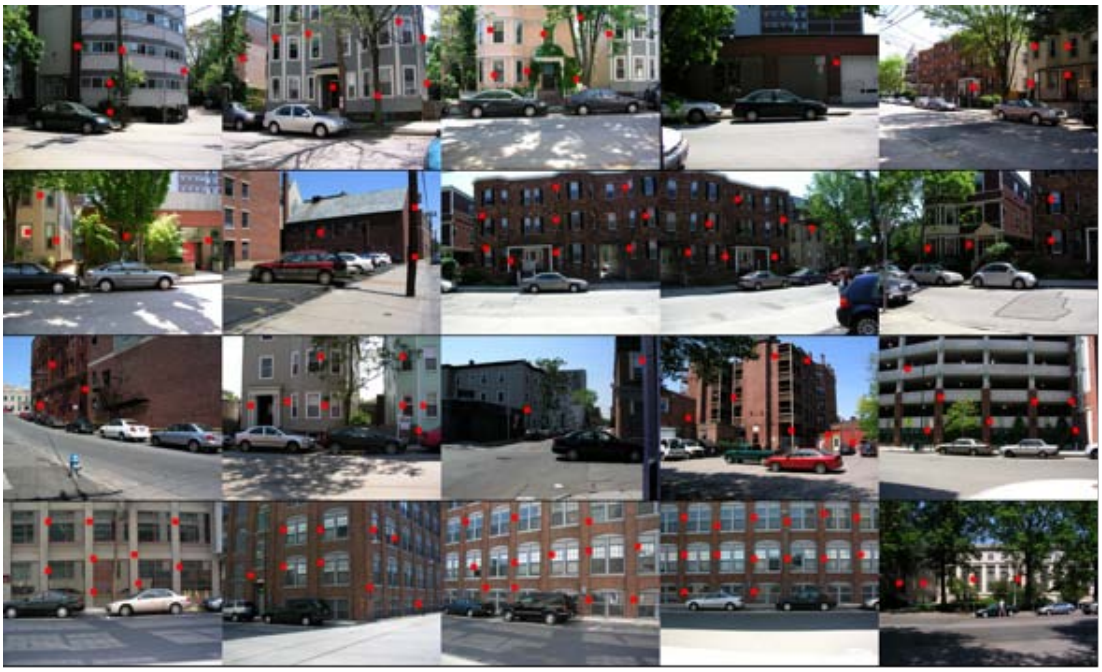


Figure 5.10: Instantiations of one of the first level rules learnt from the street scene images (from the LabelMe dataset). The repeated multiple instantiations of the same rule to explains a variety of windows on the buildings can be seen.

Figure 5.11: The result of using PLSA on the street scene images, with K = 5 topics. Each row corresponds to a topic, displaying the images which were assigned to that topic, along with the features in the image (document) that were assigned the highest probability for that topic.

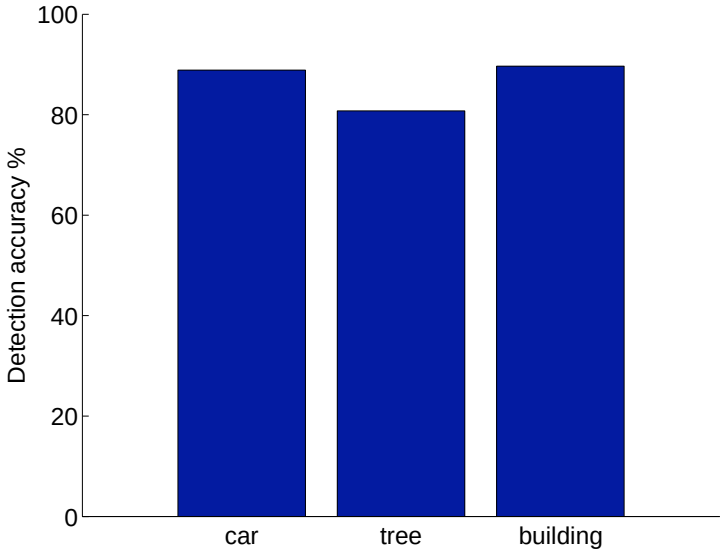


Figure 5.12: The detection accuracy of these categories i.e. the proportion of the objects that their corresponding rules fired on.

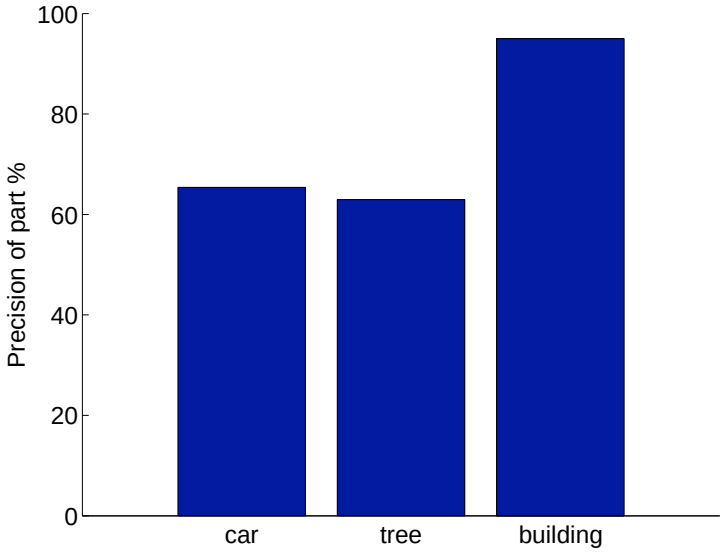


Figure 5.13: The precision of the parts associated with each of these categories i.e. the proportion of parts that fired on the objects

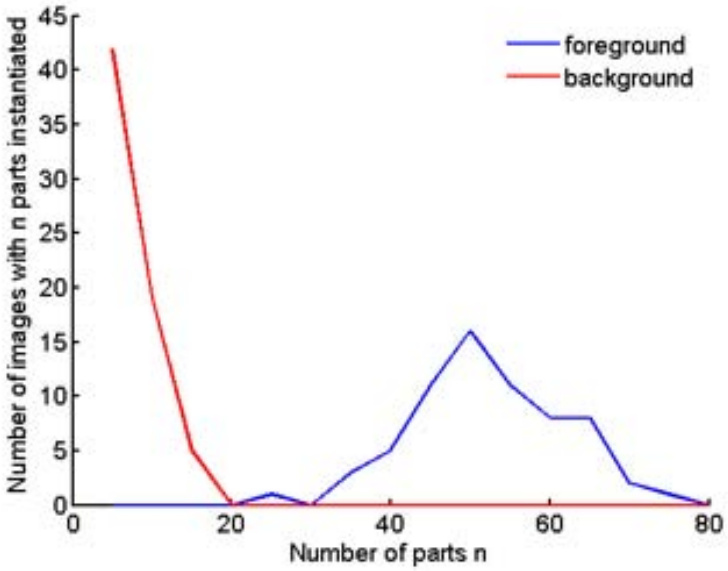


Figure 5.14: The number of parts learnt from street scenes (foreground) that were instantiated on background images. The rules learnt capture the spatial structures of the dataset, and not noise.

Chapter 6

Conclusions

Incorporating context in image understanding has received significant attention in recent works. Contextual information is often learnt in a supervised manner and utilized to enhance performance of higher level tasks such as object recognition or detection. In this thesis, we took a closer look at the role of context in image understanding. Specifically, we asked three questions. First: *When* is context really helpful? We found, through computer vision experiments as well as human studies, that context provides improvements in recognition performances only when the appearance information is weak (such as in low resolution images or in the presence of occlusion). Second: *For what* tasks can contextual information be leveraged? We showed that apart from high-level tasks of recognition and detection, contextual information can be effectively leveraged for low level tasks as well, such as identifying salient or representative patches in an image. Lastly, *How* can context be learnt? Or alternatively, how much contextual information can be extracted in an unsupervised manner? We proposed a unified hierarchical representation for contextual interactions or spatial patterns among visual entities at all levels, from low-level features to parts of objects, objects, groups of objects and ultimately the entire scene. We presented results of our approach on a variety

of datasets such as object categories, street scenes and natural scene images.

6.1 Future Work

The following are different directions I would like to pursue in the future using my previous work as a stepping stone, via collaborations with researchers in robotics, human perception and artificial intelligence.

6.1.1 Extension to video

I would like to investigate a similar approach in videos for unsupervised learning of hierarchical spatio-temporal patterns. Meaningful patterns among low-level spatio-temporal features, mid-level actions as well as higher-level events in videos can be extracted. For example, from a collection of un-annotated videos of thefts, we can learn that the event of a theft can be characterized by a sequence of actions such as approaching, picking an object up, and swiftly walking away at certain intervals of time. Each of these actions, such as walking, can be characterized by mid-level actionlets such as extending one leg at a time, and each of these can further be characterized by relevant low-level spatio-temporal features. Although it may seem like a direct application of the above algorithm developed for images to video, it opens up a new realm of possible applications such as action recognition, event recognition and anomaly detection.

6.1.2 Incorporating other sources of information

So far my current work has relied mostly on appearance and location information. 3D cues such as depth and occlusion information can also be incorporated.

This would allow us to leverage a new set of statistics to better learn the spatial patterns. Moreover, incorporating 3D information would allow for more robust estimates of certain statistics such as location and scale that are currently estimated in 2D.

It would be interesting to explore other forms of information such as audio associated with video, and perhaps learn a meaningful hierarchy on the audio signals, so that the audio and visual hierarchies enhance each other. Alternatively, the audio entities can be incorporated in the visual hierarchy. Thus events and actions can now be characterized with the visual spatio-temporal patterns as well as the audio patterns and the temporal interactions among these audio patterns. For example, the event of a first-time meeting can be characterized as the action of extending a hand, shaking someones hand, saying nice to meet you, and pulling your hand back. Each of these can further be characterized by the corresponding lower-level entities. This would allow for a very rich and multi-modal hierarchical description of an event.

6.1.3 Understanding human abilities

Another interesting avenue that I wish to explore is to perform human studies to truly understand the extent to which visual cues alone allow humans to understand such interactions in a new visual world that we create where there are no biases of functionality cues or common knowledge. For instance, we could create a synthetic scene, with synthetic objects, and assume a certain hierarchical representation. We would simulate the interactions among these synthetic objects in the scene, through their relative locations for instance, according to this as-

sumed hierarchy. Human subjects can be shown several images of such a scene and asked to describe the hierarchy they understand from these images. It would be interesting to observe the performance of humans in this synthetic world when their biases of the real world are not relevant anymore.

Questions such as how many images does a human need to observe to pick up the hierarchy, or how long humans need to view the images can be studied. It would be challenging to set up such an experiment with video, while ensuring that the objects, their actions and the scene do not remind humans of any object or event in the real world. This would lead to several interesting human experimental design and calibration questions. Can we quantify how closely a human subject associates a certain synthetic object/action to a real world object/action? Are we, as humans, even capable of designing an object/action that doesnt resemble something in the real world?

Such experiments would potentially serve two purposes: provide an empirical bound on how much information we can expect a machine to learn in an unsupervised setting with just visual cues and secondly, perhaps provide new insights to the human perception community.

6.1.4 Building a system

I would like to develop applications and a system that uses such an unsupervised algorithm in the real world. For instance, a robot can explore and observes a given scene, collect all the visual data, analyze it with this unsupervised algorithm, understands the interactions among the objects in the scene, and in turn exploit this understanding for any task at hand such as object detection, object recognition

and anomaly detection. Apart from the task of building and programming the robot, interesting path planning research questions could arise: what parts of the scene are already well understood? What parts of the scene should be further explored? This leads to interesting information theoretic questions of what it means to have understood a scene well. Ways to incorporate some supervision in the system if the user so prefers would also be explored for the system to be more realistic and usable.

Appendix A

Related Publications

- D. Parikh, and T. Chen. Unsupervised Modeling of Objects and their Hierarchical Contextual Interactions. EURASIP Journal on Image and Video Processing, Special Issue on Patches in Vision, 2008
- D. Parikh, L. Zitnick and T. Chen. Unsupervised Learning of Hierarchical Spatial Structures in Images. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009 (to appear)
- D. Parikh, L. Zitnick, and T. Chen. Determining Patch Saliency Using Low-Level Context, European Conference on Computer Vision (ECCV), 2008
- D. Parikh, L. Zitnick, and T. Chen. From Appearance to Context-Based Recognition: Dense Labeling in Small Images, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008
- D. Parikh, and T. Chen. Unsupervised Identification of Multiple Objects of Interest from Multiple Images: dISCOVER, Asian Conference in Computer

Vision (ACCV), 2007

- D. Parikh, and T. Chen. Hierarchical Semantics of Objects (hSOs), IEEE International Conference in Computer Vision (ICCV), 2007

- D.Parikh, and T. Chen. Unsupervised Learning of Hierarchical Semantics of Objects (hSOs). Beyond Patches Workshop, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007. (Best Paper Award)

References

[1] Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal on Computer Vision (IJCV) (2001) xi, 48, 49, 71, 106, 121

[2] Pascal01: http://pascallin.ecs.soton.ac.uk/challenges/VOC/ xi, 48, 49

[3] Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2003) 5, 67, 104, 107, 108

[4] Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. International Journal on Computer Vision (IJCV) (2005) 5, 104, 105, 107, 114

[5] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Generative-Model Based Vision (2004) 5, 106, 117, 120

[6] Griffin, G., Holub, A., Perona, P.: The caltech-256 object category dataset. California Institute of Technology, Technical Report (2007) 5

[7] Rabinovich, A., Vedaldi, A., Galleguillos, C., E.Wiewiora, Belongie, S.: Objects in context. International Conference in Computer Vision (ICCV) (2007) 5, 6, 7, 10, 14, 23, 29, 41, 104, 107

[8] Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 5, 6, 41, 70, 104, 107

[9] Torralba, A., Murphy, K., Freeman, W.: Contextual models for object detection using boosted random fields. Neural Information Processing Systems Conference (NIPS) (2005) 5, 6, 41, 70, 104

[10] Torralba, A., Sinha, P.: Statistical context priming for object detection. International Conference in Computer Vision (ICCV) (2001) 5, 6, 41, 70

[11] Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the trees: a graphical model relating features, objects, and scenes. Neural Information Processing Systems Conference (NIPS) (2003) 5, 6, 41, 70, 104, 107

[12] He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale conditional random fields for image labeling. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2004) 5, 6, 12, 23, 70

[13] Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: joint appearance, shape and context modeling for multi-class object recognition and

segmentation. European Conference in Computer Vision (ECCV) (2006) 5, 6, 11, 15, 23, 55

[14] Carbonetto, P., Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. European Conference in Computer Vision (ECCV) (2004) 5, 6

[15] Fink, M., Perona, P.: Mutual boosting for contextual inference. Neural Information Processing Systems Conference (NIPS) (2003) 5, 6

[16] Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. International Conference in Computer Vision (ICCV) (2005) 5, 6, 70, 104, 107

[17] Singhal, A., Luo, J., Zhu, W.: Probabilistic spatial context models for scene content understanding. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2003) 5, 7, 41, 71, 104, 107

[18] Bose, B., Grimson, E.: Improving object classification in far-field video. European Conference in Computer Vision (ECCV) (2004) 5, 6, 41

[19] Torralba, A., Murphy, K., Freeman, W., Rubin, M.: Context-based vision system for place and object recognition. Massachusetts Institute of Technology (MIT), AI Memo (2003) 5, 6, 41, 70

[20] Parikh, D., Chen, T.: Hierarchical semantics of objects (hsos). International Conference in Computer Vision (ICCV) (2007) 64, 104, 105, 107

[21] Torralba, A., Fergus, R., Freeman, W.: Tiny images. Massachusetts Institute of Technology (MIT), Technical Report (2007) 8, 9, 10, 16, 17

[22] Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. International Conference in Computer Vision (ICCV) (2003) 8

[23] MSRC-Dataset: http://research.microsoft.com/vision/cambridge/recognition/ 9, 15

[24] Corel-Dataset: http://www.cs.toronto.edu/~hexm/label.htm 9, 15

[25] Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. British Machine Vision Conference (BMVC) (2007) 10

[26] Rabinovich, A., Vedaldi, A., Belongie, S.: Does image segmentation improve object categorization? University of California Sand Diego, Technical Report (2007) 10

[27] Shotton, J.: http://jamie.shotton.org/work/code.html 11

[28] Meltzer, T.: http://www.cs.huji.ac.il/~talyam/inference.html 14, 97

[29] Bachmann, T.: Identification of spatially queatized tachistoscopic images of faces: how many pixels does it take to carry identity? European Journal of Cognitive Psychology (1991) 17

[30] Harmon, L., Julesz, B.: Masking in visual recognition: effects of two-dimensional noise. Science (1973) 17

[31] Oliva, A.: Gist of the scene. Neurobiology of Attention (2005) 17

[32] Oliva, A., Schyns, P.: Diagnostic colors mediate scene recognition. Cognitive Psychology (1976) 17

[33] Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal on Computer Vision (IJCV) (2004) 22, 121

[34] Yang, L., Meer, P., Foran, D.: Multiple class segmentation using a unified framework over mean-shift patches. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 23

[35] Verbeek, J., Triggs, B.: Region classification with markov field aspect models. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 23

[36] He, X., Zemel, R., Ray, D.: Learning and incorporating top-down cues in image segmentation. European Conference in Computer Vision (ECCV) (2006) 23

[37] Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: a database and web-based tool for image annotation. Massachusetts Institute of Technology (MIT), AI Memo (2005) 29, 106, 122

[38] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal on Computer Vision (IJCV) (2004) 38, 40, 41, 42, 47, 50, 72, 117, 124

[39] Harris, C., Stephens, M.: A combined corner and edge detector. Alvey Vision Conference (1988) 38, 40, 47

[40] Kadir, T., Brady, M.: Saliency, scale and image description. International Journal on Computer Vision (IJCV) (2001) 38, 40, 41

[41] Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal on Computer Vision (IJCV) (2004) 38, 40

[42] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. British Machine Vision Conference (BMVC) (2002) 38, 40

[43] Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal on Computer Vision (IJCV) (2000) 38, 40

[44] Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. European Conference in Computer Vision (ECCV) (2006) 38, 39, 40, 41, 49, 51, 52

[45] Moosmann, F., Larlus, D., Jurie, F.: Learning saliency maps for object categorization. European Conference in Computer Vision (ECCV), International Workshop on The Representation and Use of Prior Knowledge in Vision (2006) 38, 40

[46] Walker, K., Cootes, T., Taylor, C.: Locating salient object features. British Machine Vision Conference (BMVC) (1998) 38

[47] Fritz, G., Seifert, C., Paletta, L., Bischof, H.: Entropy based saliency maps for object recognition. Early Cognitive Vision Workshop (2004) 38

[48] Serre, T., Riesenhuber, M., Louie, J., Poggio, T.: On the role of object-specific features for real world object recognition in biological vision. British Machine Vision Conference (BMVC) (2002) 38

[49] Vidal-Naquet, M., Ullman, S.: Object recognition with informative features and linear classification. International Conference in Computer Vision (ICCV) (2003) 38, 41

[50] Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal on Computer Vision (IJCV) (2001) 39, 41

[51] Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. International Conference in Computer Vision (ICCV) (2003) 39

[52] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. European Conference in Computer Vision (ECCV), workshop on Statistical Learning in Computer Vision (2004) 39, 40, 66

[53] Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. International Conference in Computer Vision (ICCV) (2005) 39, 41

[54] Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from googles image search. International Conference in Computer Vision (ICCV) (2005) 39, 40, 66

[55] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. International Conference in Computer Vision (ICCV) (2003) 39, 40

[56] Sivic, J., Russell, B., Efros, A.A., Zisserman, A., Freeman, B.: Discovering objects and their location in images. International Conference in Computer Vision (ICCV) (2005) 39, 40, 66, 67, 104, 107, 108, 117

[57] Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. International Conference in Computer Vision (ICCV) (2005) 40

[58] Ye, Y., Tsotsos, J.K.: Where to look next in 3d object search. IEEE International Symposium for Computer Vision (1995) 40

[59] Viola, P., Jones, M.: Robust real-time object detection. International Journal on Computer Vision (IJCV) (2001) 40

[60] Grauman, K., Darrell, T.: Efficient image matching with distributions of local invariant features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 40

[61] Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 40, 66, 67, 71

[62] Agarwal, A., Triggs, B.: Hyperfeatures multilevel local coding for visual recognition. European Conference in Computer Vision (ECCV) (2006) 41

[63] Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology (1980) 41

[64] Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (1998) 41

[65] Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiology (1985) 41

[66] Sebe, N., Lew, M.: Comparing salient point detectors. Pattern Recognition Letters (2003) 41

[67] Hall, D., Leibe, B., Schiele, B.: Saliency of interest points under scale changes. BNVC (2002) 41

[68] Walther, D., Rutishauser, U., Koch, C., Perona, P.: On the usefulness of attention for object recognition. European Conference in Computer Vision (ECCV) (2004) 41

[69] Parikh, D., Zitnick, C.L., Chen, T.: From appearance to context-based recognition: Dense labeling in small images. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 41, 98, 104, 107

[70] Torralba, A.: Outdoor scene category dataset, http://people.csail.mit.edu/torralba/code/spatialenvelope/ 49, 51

[71] Parikh, D., Chen, T.: Unsupervised identification of multiple objects of interest from multiple images: discover. Asian Conference in Computer Vision (ACCV) (2007) 64

[72] Fitzgibbon, A., Zisserman, A.: On affine invariant clustering and automatic cast listing in movies. European Conference in Computer Vision (ECCV) (2002) 66

[73] Sivic, J., Zisserman, A.: Video data mining using configurations of viewpoint invariant regions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2004) 66

[74] Berg, T., Berg, A., Edwards, J., White, R., Teh, Y., Learned-Miller, E., Forsyth, D.: Names and faces in the news. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2004) 66

[75] Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning (2001) 66, 67

[76] Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research (2003) 66

[77] Quelhas, P., Monay, F., Odobez, J.M., Gatica-Perez, D., Tuytelaars, T., Gool, L.V.: Modeling scenes with local descriptors and latent aspects. International Conference in Computer Vision (ICCV) (2005) 66, 67

[78] Leordeanu, M., Collins, M.: Unsupervised learning of object models from video sequences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (200r) 66

[79] Liu, D., Chen, T.: Semantic-shift for unsupervised object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Beyond Patches (2006) 66

[80] Li, Y., Wang, W., Gao, W.: A robust approach for object recognition. Advances in Multimedia Processing, Pacific Rim Conference on Multimedia (PCM) (2006) 66

[81] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 66

[82] Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 66

[83] Marszałek, M., Schmid, C.: Spatial weighting for bag-of-features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 67

[84] Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. European Conference in Computer Vision (ECCV) (2004) 67

[85] Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. European Conference in Computer Vision (ECCV) (2000) 67

[86] Sudderth, E., Torralba, A., Freeman, W., A.Wilsky: Learning hierarchical models of scenes, objects, and parts. International Conference in Computer Vision (ICCV) (2005) 68, 104, 107

[87] Wang, G., Zhang, Y., Fei-Fei, L.: Using dependent regions for object categorization in a generative framework. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 68

[88] Marr, D., Nishihara, H.: Representation and recognition of the spatial organization of three dimensional structure. Proceedings of the Royal Society of London B (1978) 68

[89] Biederman, I.: Human image understanding: recent research and a theory. Computer Vision, Graphics and Image Processing (CVGIP) (1985) 68

[90] Bienenstock, E., Geman, S., Potter, D.: Compositionality, mdl priors, and object recognition. Neural Information Processing Systems Conference (NIPS) (1997) 68

[91] Levinshtein, A., Sminchisescu, C., Dickinson, S.: Learning hierarchical shape models from examples. International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR) (2005) 68

[92] Bouchard, G., Triggs, W.: Hierarchical part-based visual object categorization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 68

[93] Jin, Y., Geman, S.: Context and hierarchy in a probabilistic image model. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 68, 104, 107

[94] Fidler, S., Berginc, G., Leonardis, A.: Hierarchical statistical learning of generic parts of object structure. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 68

[95] Siskind, J., Sherman, J., Pollak, I., Harper, M., Bouman, C.: Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (to appear) 68

[96] Forsyth, D., Mundy, J., Zisserman, A., Rothwell, C.: Using global consistency to recognise euclidean objects with an uncalibrated camera. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1994) 70

[97] Torralba, A., Oliva, A.: Depth estimation from image structure. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2002) 70

[98] Williams, C., Adams, N.: Dts: Dynamic trees. Neural Information Processing Systems Conference (NIPS) (1999) 70

[99] Hinton, G., Ghahramani, Z., Teh, Y.: Learning to parse images. Neural Information Processing Systems Conference (NIPS) (2000) 70

[100] Storkey, A., Williams, C.: Image modelling with position encoding dynamic trees. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2003) 70

[101] Tu, Z., Chen, X., Yuille, A., Zhu, S.: Image parsing: unifying segmentation, detection, and recognition. International Journal on Computer Vision (IJCV) (2005) 70

[102] Marszałek, M., C.Schmid: Semantic hierarchies for visual object recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 70

[103] Vogel, J., Schiele, B.: A semantic typicality measure for natural scene categorization. Pattern Recognition Symposium, DAGM (2004) 71

[104] Belongie, S., Malik, J., Puzicha, J.: Shape context: a new descriptor for shape matching and object recognition. Neural Information Processing Systems Conference (NIPS) (2000) 72

[105] Berg, A., Malik, J.: Geometric blur for template matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2001) 72

[106] Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. International Conference in Computer Vision (ICCV) (2005) 75

[107] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2000) 79, 117

[108] Shi, J.: http://www.cis.upenn.edu/~jshi/software/ 79

[109] Nakazato, M., Manola, L., Huang, T.: Imagegrouper: search, annotate and organize image by groups. Recent Advances in Visual Information Systems (VISual) (2002) 89

[110] Crandall, D., Felzenszwalb, P., , Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 104, 107, 108

[111] Leordeanu, M., Hebert, M., Sukthankar, R.: Category recognition from pairwise interactions of simple features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 104, 108

[112] Karlinsky, L., Dinerstein, M., Levi, D., Ullman, S.: Unsupervised classification and part localization by consistency amplification. European Conference in Computer Vision (ECCV) (2008) 104, 108

[113] Liu, D., Hua, G., Viola, P., Chen, T.: Integrated feature selection and higher-order spatial feature extraction for object categorization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 104

[114] Nakazato, M., Manola, L., Huang, T. Syntactic Pattern Recognition and Applications (1982) 104, 107

[115] Ullman, S.: Visual routine. Cognition (1984) 104, 107

[116] Zhu, S., Mumford, D.: Quest for a stochastic grammar of images. Foundations and Trends in Computer Graphics and Vision (2007) 104, 107, 108

[117] Hwang, F., Richards, D., Winter, P.: The steiner tree problem. (1992) 105, 112

[118] Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S.: Approximation algorithms for directed steiner problems. Symposium on Discrete Algorithms (1998) 106, 113, 124

[119] Hanson, A., Riseman, E.: Visions: a computer system for interpreting scenes. Computer Vision Systems (1978) 107

[120] Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2007) 107

[121] Todorovic, S., Ahuja, N.: Learning subcategory relevances to category recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 107, 108

[122] Zhu, L., Lin, C., Huang, H., Chen, Y., Yuille, A.: Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. European Conference on Computer Vision (ECCV) (2008) 107, 108

[123] Fidler, S., Berginc, G., Leonardis, A.: Hierarchical statistical learning of generic parts of object structure. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 107

[124] Sivic, J., Russell, B., Zisserman, A., Freeman, W., Efros., A.: Unsupervised discovery of visual object class hierarchies. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 108

[125] Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Describing visual scenes using transformed objects and parts. International Journal on Computer Vision (IJCV) (2008) 108

[126] Epshtein, B., Ullman, S.: Feature hierarchies for object classification. International Conference in Computer Vision (ICCV) (2005) 108

[127] Fidler, S., Leonardis, A.: Towards scalable representations of object categories: Learning a hierarchy of parts. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 108

[128] Zhu, L., Chen, Y., Lu, Y., Lin, C., Yuille, A.: Max margin and/or graph learning for parsing the human body. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 108

[129] Han, F., Zhu, S.C.: Bottom-up/top-down image parsing by attribute graph grammar. International Conference in Computer Vision (ICCV) (2005) 108

[130] Liu, D., Chen, T.: Unsupervised image categorization and object localization using topic models and correspondences between images. International Conference in Computer Vision (ICCV) (2007) 108

[131] Kim, G., Faloutsos, C., Hebert, M.: Unsupervised modeling of object categories using link analysis techniques. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008) 108, 120