Universitat de Girona

IMAGE CLASSIFICATION FOR A LARGE NUMBER OF OBJECT CATEGORIES

Anna BOSCH I RUÉ

ISBN: 978-84-691-1539-8
Dipòsit legal: GI-1407-2007
http://hdl.handle.net/10803/7884

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.

PhD Thesis

IMAGE CLASSIFICATION FOR LARGE NUMBER OF OBJECT CATEGORIES

Anna Bosch Rué

Universitat
de Girona

Department of Electronics, Informatics and Automation,
University of Girona.

Supervisors
Dr. Xavier Muñoz Pujol
Prof. Andrew Zisserman

Als qui que fan els seus somnis realitat

Agraïments

Hi ha dues persones sense les quals aquesta tesi no hauria estat possible de cap manera. M'agradaria donar les gràcies als meus dos supervisors: al Dr. Xavier Muñoz i al *Professor* Andrew Zisserman pel seu entusiasme i fer de guies de la tesi. Gràcies a en Xevi per ser director i amic i a l'Andrew per haver confiat en mi i haver-me donat una oportunitat.

Moltes gràcies també a tots els autors que han fet públiques les seves bases de dades d'imatges, sense això no hauria pogut treballar, gràcies doncs a en Greg Griffin, la Lana Lazebnik, la Fei Fei Li, l'Aude Oliva, en Pietro Perona, en Jean Ponce, en Bert Schiele, la Cordelia Schmid, en Josef Scivic, l'Antonio Torralba i la Julia Vogel. Gràcies a en Rob Fergus i en Josef Sivic pels seus comentaris i suggerències. I com no, gràcies a tots els revisors anònims que amb els seus comentaris també han contribuït a que aquest treball sigui una mica millor, en especial moltes gràcies a la Tinne Tuytelaars i la Cordelia Schmid per acceptar ser les dues revisores de la tesi.

Gràcies en general a tota la gent de la Universitat de Girona que d'una manera o altra han contribuït a la tesi. M'agradaria destacar en Jordi Freixenet, en Quim Salvi i en Joan Martí per haver-me introduït en el món de la visió per computador. Gràcies a l'Arnau per estar al Lab i fer que no sigui tan avorrit! a en Xavier Lladó per haver revisat la tesi i sobretot per fer-me riure molt en els congressos que hem coincidit i a en Robert Martí per corregir-me les faltes angleses d'algun treball. Gràcies també a en Tudor, la Marina, en Jordi, en Josep, l'Olivier etc, pels moments de *piti* compartits.

Unes gràcies molt especials van per la Tere, amb qui vaig compartir molt bons moments durant la primera estada a Oxford, la veritat, no sé què hauria fet sense ella! Agrair també a tota la gent d'Oxford que van fer que les meves estades allà fossin agradables.

Gràcies a la meva ”family-in-law”, sobretot a en Toni i la Carme perquè sempre s'han interessat pel que faig. Gràcies als meus amics, en especial a l'Andrea i en Xevi per fer-me sentir tan be quan estic amb ells i per preocupar-se per mi. A la Bet i en Jordi per les escalades i caminades que hem fet junts, i per ser dues persones amb les qui parlar és molt fàcil.

Vull donar les gràcies als meus pares per haver sabut educar una filla tan rebel (tot i que no la k més, oi?) i per tota la paciència que han tingut i el suport que m'han donat. A la meva mare per fer-me entrar en raó i fer-me veure que les lletres no eren lo meu. Al meu pare per sempre donar-me un cop de mà sense demanar res a canvi, per sempre preocupar-se per tot, perquè sempre m'ha donat el millor i mai m'ha faltat res, per sempre pensar en les seves filles abans que en ell. Gràcies a la Sara per ser germana, amiga i confident. I les meves nenes! La Neus i L'Èlia, tan diferents com la nit i el dia, però que me les estimo moltíssim a les dues tal i com elles són. I sobretot perquè inconscientment he après d'elles moltes més coses de les que s'imaginen. Unes gràcies molt especials a la meva Padrineta, perquè sempre té un somriure a la boca, perquè si hi hagués més persones com ella segur que aquest món seria molt millor. Gràcies també al meu avi Andreuet per recordar-me que no hi ha res que s'aconsegueixi sense esforç.

Les meves últimes gràcies van per en Robert. Perquè sempre té els nostres somnis presents, perquè sap treure el millor de mi i em fa voler millorar com a persona dia rera dia. Gràcies per sempre fer-me costat, per inspirar-me, per ajudar-me i perquè tot i la distància no hem deixat que res ens separi.

Acknowledgements

For those who do not speak catalan, this section is for you. I would like to start saying thanks to Professor Andrew Zisserman to give me the opportunity to do several research stays at the Oxford University. This thesis would not have been possible without his guidance and enthusiasm.

This thesis would neither be possible without all the public image datasets that I used, so thanks to Greg Griffin, Lana Lazebnik, Fei Fei Li, Aude Oliva, Pietro Perona, Jean Ponce, Bert Schiele, Cordelia Schmid, Josef Scivic, Antonio Torralba and Julia Vogel.

I would also like to thank Rob Fergus and Josef Scivic for their comments and suggestions. I also have to thank all the anonymous reviewers who with their comments also contribute to improve this work. My special thanks to Tinne Tuytelaars and Cordelia Schmid who kindly accepted to review this thesis.

Abstract

The release of challenging data sets with ever increasing numbers of object categories is forcing the development of image representations that can cope with multiple classes and of algorithms that are efficient in training and testing. This thesis explores the problem of classifying images by the object they contain in the case of a large number of categories.

We first investigate weather the hybrid combination of a latent generative model with a discriminative classifier is beneficial for the task of weakly supervised image classification. We introduce a novel vocabulary using dense color SIFT descriptors, and then investigate classification performances by optimizing different parameters. A new way to incorporate spatial information within the hybrid system is also proposed showing that contextual information provides a strong support for image classification.

We then introduce a new shape descriptor that represents local image shape and its spatial layout, together with a spatial pyramid kernel. Shape is represented as a compact vector descriptor suitable for use in standard learning algorithms with kernels. Experimental results show that shape information has similar classification performances and sometimes outperforms those methods using only appearance information.

We also investigate how different cues of image information can be used together. We will see that shape and appearance kernels may be combined and that additional information cues increase classification performance.

Finally we provide an algorithm to automatically select the regions of interest in training. This provides a method of inhibiting background clutter and adding invariance to the object instance's position. We show that shape and appearance representation over the regions of interest together with a random forest classifier which automatically selects the best cues increases on performance and speed.

We compare our classification performance to that of previous methods using the authors' own datasets and testing protocols. We will see that the set of innovations introduced here lead for an impressive increase on performance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thousands of images are generated every day, which implies the necessity to classify, organize and access them by an easy and faster way. With the exponential growth on high-quality digital images, the need of semantic image classification is becoming increasingly important to support effective image database indexing and retrieval. Classifying images into semantic categories (e.g. coast, mountains, streets) and also classify its semantic objects (e.g. motorbikes, sky, planes) is a challenging and important problem nowadays. This chapter will first describe the thesis objectives and motivations. We will then answer why it is a challenge and what we have achieved over the last years. An outline of the thesis is finally given.

1.1 Objectives

The objective of this work is image classification. Given a set of images our objective is to classify them by the scene/object category they contain (e.g. *coast*, *forest*, *kitchen*, *cars*, *leopards*). Figure 1.1 and figure 1.2 show some images we are interested in. This problem has been the subject of many recent papers [63, 74, 76, 102, 157, 158] using specific scene classification datasets, the Pascal Visual Object Classes datasets or the Caltech-101 dataset. The release of challenging data sets with ever increasing numbers of object categories, such as the recent Caltech-256 [59], is forcing the development of image representations that can cope with multiple classes and of algorithms that are efficient in training and testing.

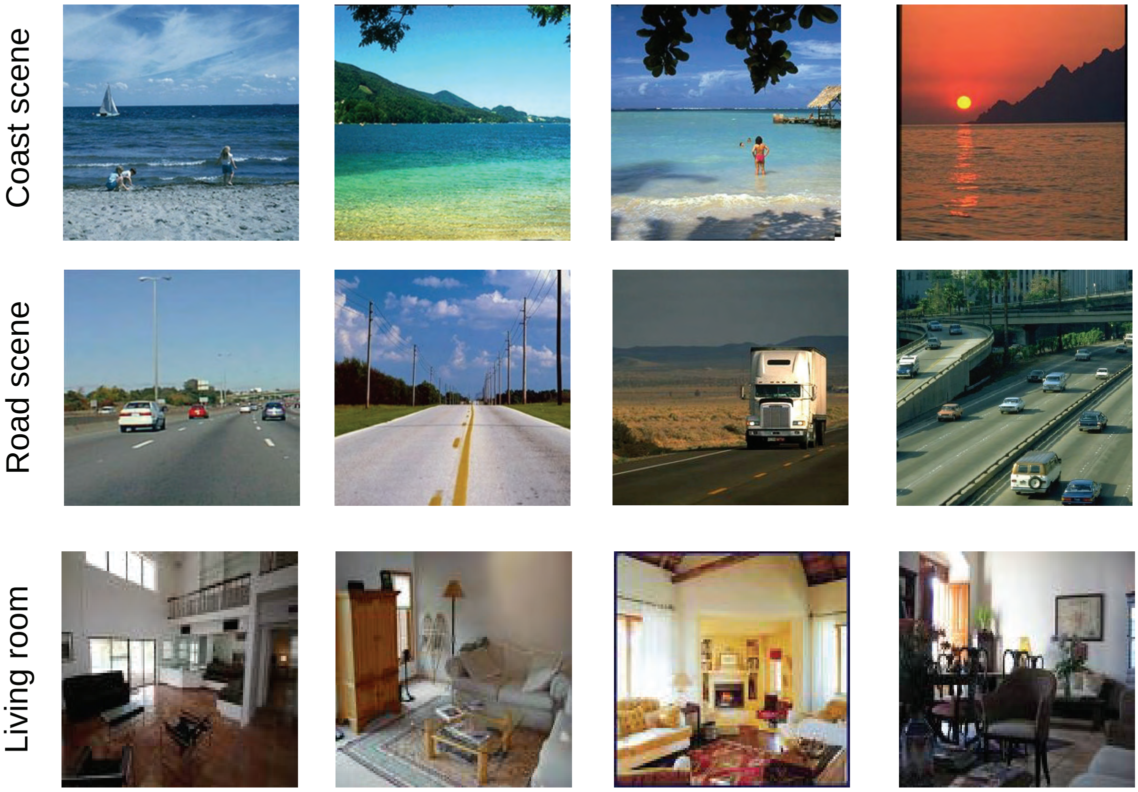


Figure 1.1: Images representing different scenes: coast, road, living room.

Here, we want to distinguish amongst a large amount of image categories (up to 256). Our goal will be to develop an image classification system reducing the amount of manual supervision required as well as reducing the computational cost to learn the classifier. We are looking for a trade-off amongst efficiency, supervision and performance. These characteristics are crucial for enabling it to function in real-world applications.

Moreover, when lots of categories are used it is not enough to use only one feature (say the shape) of the objects to distinguish amongst them. For example shape may be a good feature to distinguish between *cars* and *airplanes* but it is not good to distinguish between *horses* and *zebras*. Our goal will be to use features and combination of features which provide a discriminative image representation amongst all the categories we want to classify.

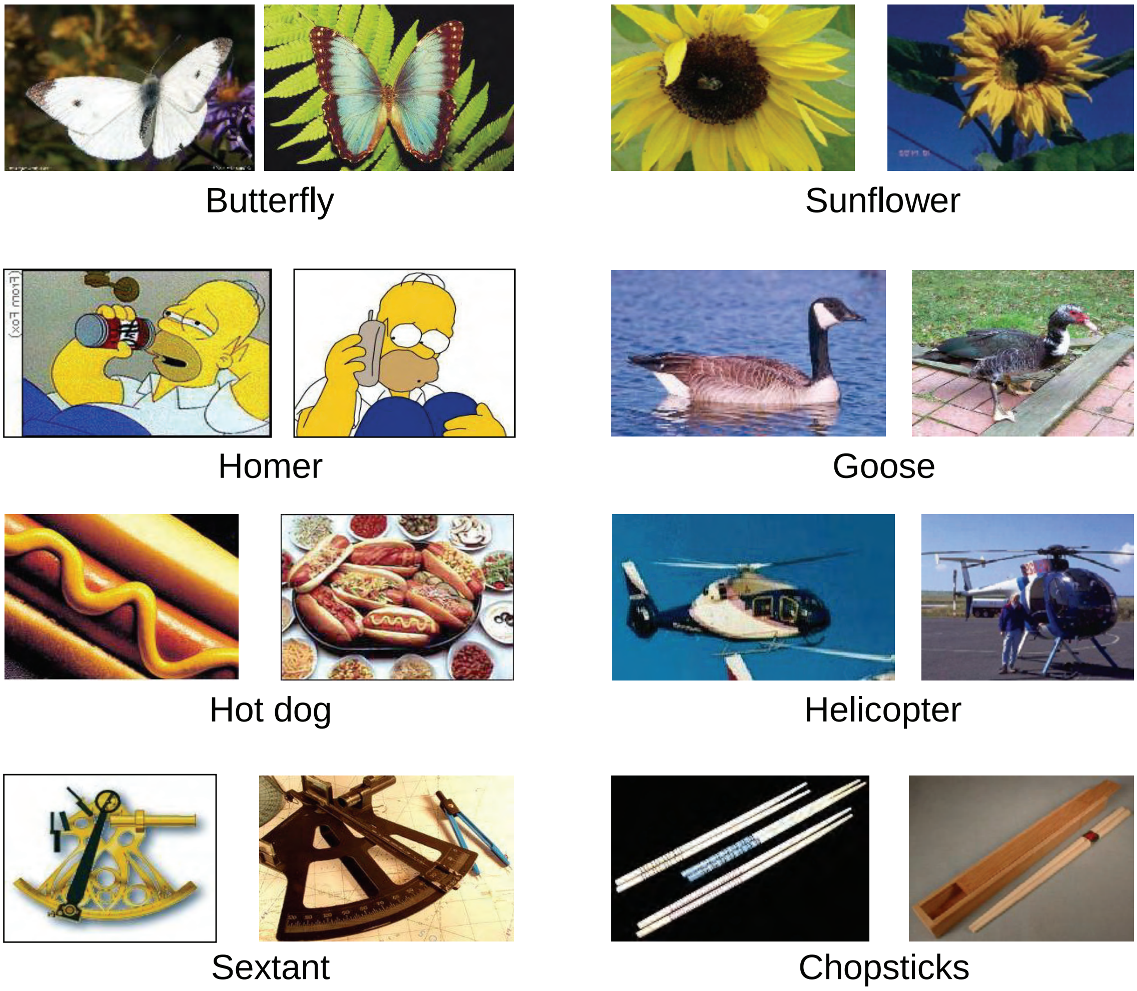


Figure 1.2: Images containing different object categories. Note that by object we mean animals, man-made objects, insects, transport etc.

1.2 Motivation

Since I bought my first digital camera I have taken thousands of images which are stored in my computer. Every time I want to see my photos from a specific travel I have to look through almost all the folders to find the ones which I am looking for. It would be easier if I could access them asking by its content! In this way I could access very fast to my sailing photos, or the photos of my young sister. And, if just by myself I generate such amount of pictures... what about expert photograph companies?

Describing images by its content can be very useful to organize and access this amount of image data generated every day. Moreover there are lots of applications that can benefit from the scene and object classification:

- **Image search**. Image search is the most direct application when people talk about image classification. In this sense, we can think about searching images in the biggest database in the world, Internet image search engines, or simply provide applications to search images in a personal computer. Nowadays the poor performance when searching images in Internet is due to their use of the image filename or surrounding HTML rather than the actual image content. However, the natural way to find images is to search visually -as humans due- using computer vision methods. Moreover, many companies have large archives of images which they wish to search in.

- **Video search**. Lots of adverts and video data have been generated during last years. People working in marketing is often interested in look for coffee adverts televised in the past years, or adverts filmed in the mountains. Nowadays all these adverts are manually annotated and stored in databases using metadata information. It would be very useful to provide techniques to access them automatically, by its content. Also producers or film directors would be interested in recover by an automatic way those shots of movies filmed near a lake, or those shots where Johnny Deep appears in the middle of the ocean.

- **Medical applications**. In the medical field also lots of images are generated every day, radiographies, ecographies etc. It would be very useful for the doctors to provide tools to access at these images faster, and not looking case by case as they do. Even though one can thing that this field is very different from the objective of the thesis we will see in Appendix B that there is a lot in common.

- **Travel guide**. With the current spread of cheap flights people travel every day more and more. Instead of having a travel guide of each country we can have a digital travel guide stored in the mobile phone and retrieve the information by taking a picture from the famous cathedral, square.

- **Video Compression**. Due to the very limited bandwidth of a number of important communication channels (e.g. wireless, underwater, low-power camera networks, etc.), video communication over such channels requires substantial compression of the video signal. One of the most promising answers to this challenge is to adopt a new compression paradigm that relies heavily in scene understanding. It would allow the compression of different objects in a scene with specific compression levels in such a way as to adjust the trade-off between space reduction and visual quality on a per-object basis. The basic idea is that important objects such as actors should retain the highest visual quality, while objects in the background can be encoded with lower quality to save bytes. Here, computer vision must help to perform automatically the task of separating a video into the objects of which it is composed.

- **Surveillance**. Fundamental systems remain relatively unintelligent requiring a person screening the image sequences, looking for suspicious people and unusual events. Advanced systems try to automatically detect this unusual events. A subject of relative importance is that related to understand crowded environments (e.g. a football stadium) detecting risk situations (e.g. fights).
- **Aerial images**. National mapping agencies spend thousands of euros each year to keep their data up to date. This tedious and time consuming process often involves a person in front of a computer screen comparing the current raster/vector map with the most recent high quality satellite image. Image classification techniques could be used to detect landscape changes with minimal human interaction
- **Robotics**. Provide eyes to a robot is maybe one of the most ambitious things in the computer vision field. In this way a completely autonomous robot specialized to recognize certain objects of interest will be able to substitute humans in dangerous situations such as underwater exploration, fireman help etc.

However there is not a clear solution for the applications above mentioned and this is why image classification is a very challenging problem.

1.3 Challenges

Many satisfactory studies on image classification have been presented from 2000 and it is a not solved problem yet. This is because maybe together with the object recognition field

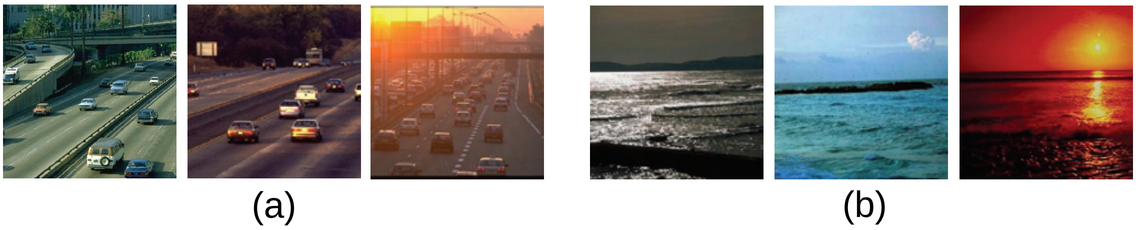


Figure 1.3: Illumination challenge. (a) Three road scenes affected by different illumination conditions; (b) three coast scenes affected by different illumination conditions.

it is one of the most challenging and ambitious problems on computer vision. Humans are able to recognize a tree even if this one is far away of if it is very close to us. The same tree has different appearances depending on the season of the year: it has no leaves in winter, brown leaves in autumn, green leaves in spring etc. and humans can recognize it in all these situations. There are lots of things that humans can due automatically but are still a challenge in computer vision. Let's go to discuss which are the major aspects we have to take into account to develop a robust image classification system:

- **Illumination**. An important thing to take into account are the illumination changes in pictures. For example, if we look at figure 1.3a we can recognize three road scenes even thought the illumination in all of them is different. Also in figure 1.3b three different coast scenes under different conditions are presented, and we are able to classify them. This is a trivial task to us, and to develop a robust system we have to consider that it has also to be able to recognize objects and scenes under different illumination conditions.

- **Intra-class variability**. Identifying instances of general object/scene classes is an extremely difficult problem, in part because of the variations among instances of many common object classes, many of which do not afford precise definitions. A coast scene can come in different ways: a paradisiac coast, a cliff coast or a coast with just water (see figure 1.4a). Also a *cormorant* can appear in different positions, in groups, on the floor, in the water as it is shown in figure 1.4b. That means we need an approach that can generalize across all possible instances of certain categories.

- **Inter-class variability**. Related to the intra-class variability problem, another major difficulty is the inter-class variability within the model. We do not want to

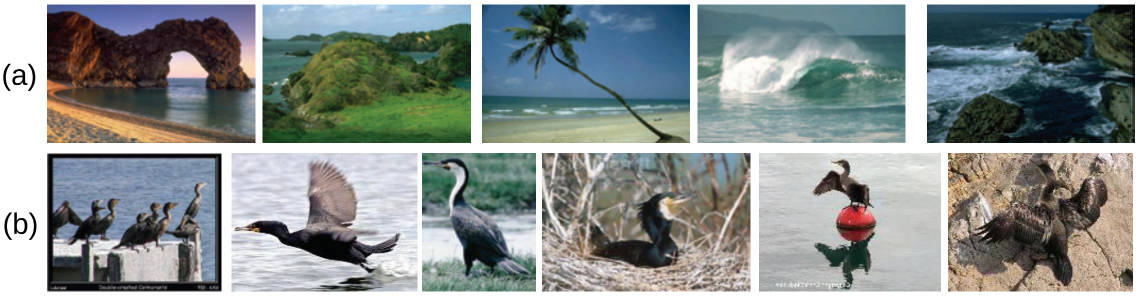


Figure 1.4: Intra-class variation problem. (a) Different coast scenes which present high intra-class variation; (b) different cormorant images which present high intra-class variation.

confuse between scenes of different categories that are quite similar. For example the forest and river scene are not labelled as the same category and we can see in figure 1.5a-b that would be easily confused. Also the *harpsichord* and *grand piano* in figure 1.5c-d can be easily confused, they have a very similar appearance and shape.

- **Scale invariance**. This is also an important thing to take into account for the scene classification problem. We can have images with a mountain in front of us, or images with a mountain far away and in both cases it is a mountain scene that the system must classify. We can also have some objects (e.g a *billiard*) which appear at different scale in the images. Figure 1.6 shows some examples related to this problem.

- **Others**. Rotations, occlusions and view point variations should also be taken into account.

A part from the above mentioned problems, for the scene classification task there are other factors related to the human perception that we would like to comment: the ambiguities and the subjectivity of the viewer. The obtainable classification accuracies depend strongly on the consistency and accuracy of the manual annotations, and sometimes annotation ambiguities are unavoidable. For example, the annotation of *mountains* and *open country* is quite challenging. Imagine an image with *fields* and *snow hills* in the far distance: is it *open country* or *mountain*? Vogel and Schiele [149] analyzed in detail the ambiguities between scene categories, showing that there is a semantic transition between

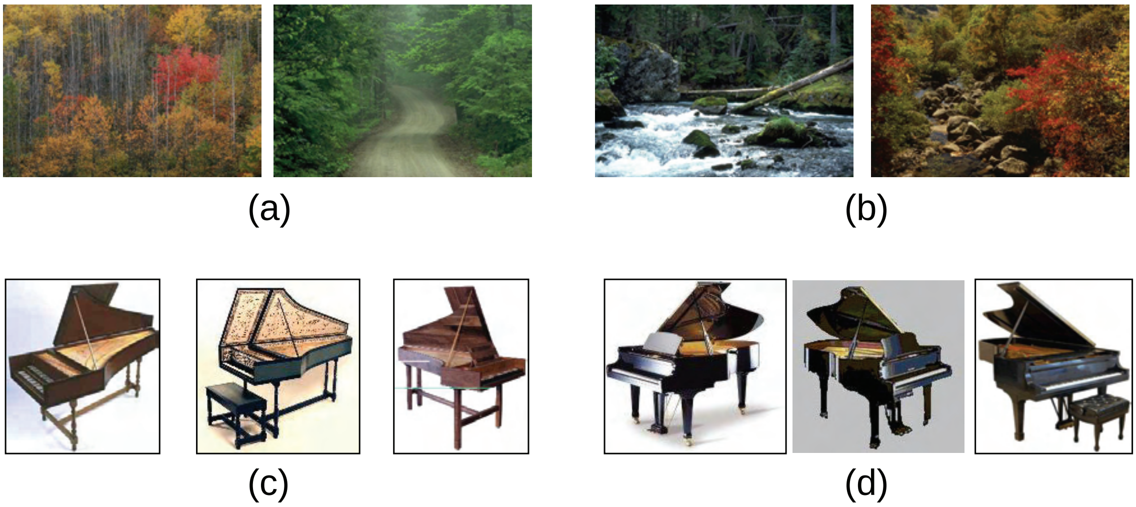


Figure 1.5: Inter-class variation problem. Two forest images (a) that could be easily confused by the two river (b) images. Three harpsichord images (c) very similar with the grand piano images (d).

categories. Their experiments with human subjects showed that many images cannot be clearly assigned to one category.

1.4 Contributions

The contribution of this thesis can be divided into six main themes, summarized below. Along the thesis we will show that these contributions allow us to achieve superior classification accuracy to recent publications, in all cases using the authors' own datasets and testing protocols. A more detailed account of contributions and how they affect the final performance will be discussed in section 8.1.

Image representation

Dense features computed over a regular grid with overlapping patches are used to represent the images. Using sparse features, as in [116, 128], the only information is where a Harris detector fires and, especially for natural images, this is a very impoverished representation. Therefore dense features provides a more robust representation when working with natural outdoor scenes. In previous work one patch is used to represent each pixel.

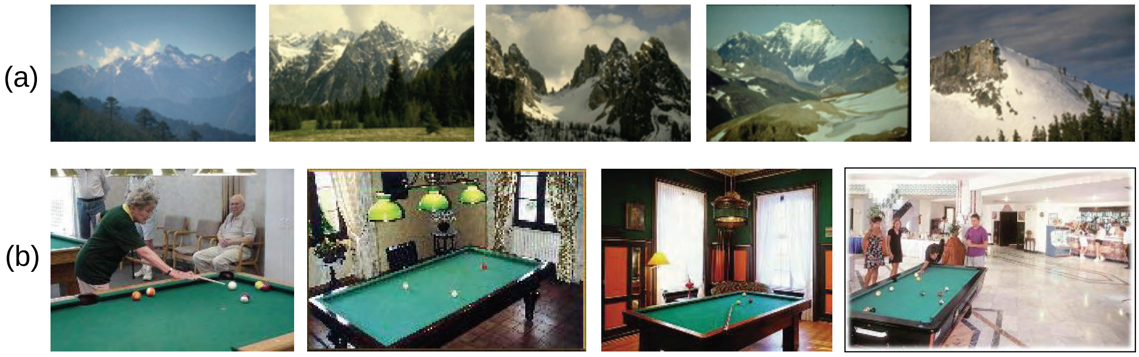


Figure 1.6: Scale variation problem. (a) Mountain scenes from different scales; (b) billiard images from different scales.

We propose to use different patches of different sizes to represent each pixel and deal with possible scale variations. In order to describe these regions most of the system use SIFT [84] over gray level images. SIFT features have been demonstrated to be one of the best when working with object recognition and scene classification [94] as well as color information [54]. So that we propose a novel descriptor which integrates color and SIFT features.

Hybrid generative/discriminative approach with spatial information

We present a weakly-supervised method for scene classification based on a dimensionality reduction using latent generative models. First we use a latent model based on probabilistic Latent Semantic Analysis (pLSA) to discover objects in images by an unsupervised way; second we represent each image by a vector, and train a multi-way classifier on these vectors (supervised step). pLSA has been previously used by Sivic *et al.* [128] however there is a main difference compared to our proposal. In [128], they classify objects as a single pLSA topic, whereas we classify scenes as a combination of pLSA topics (as scenes are composed of multiple “objects”). As a result of this, their model is completely unsupervised whereas ours is a combination of unsupervised and supervised.

Moreover pLSA makes no use of location information in the image, relying solely on the appearance of a set of regions extracted from the image. Fergus *et al.* [44] extended the pLSA model to include location information in a scale and translation invariant manner (TSI-pLSA). We extended the pLSA framework to incorporate spatial information by using a Spatial Pyramid (SP) representation [76] which is able to learn both appearance and location at different pyramid levels. In this case the image is represented by a Pyramid Histogram Of visual Words (PHOW).

A new shape descriptor

We explore how the spatial distribution of shape can benefit recognition. In essence, we wish to assess how well an exemplar image matches (the shape of) another image. To this end, we extend the spatial pyramid method of Lazebnik *et al.* [76] in two ways: the first is to represent shape in the form of edges, replacing the use of quantized appearance patches (visual words). The second extension is to learn a class specific weighting for the levels of the hierarchical spatial histogram [57, 76]. This captures the intuition that some classes are very geometrically constrained (such as a stop sign) whilst others have greater geometric variability (e.g. dolphins, boats). This descriptor is termed PHOG (for Pyramid of Histograms of Orientation Gradients)

Merging cues

Having developed the PHOG descriptor we then introduce kernels, suitable for an SVM classifier, that combine both appearance (visual words) and edge (PHOG) descriptors. This is a form of feature combination and selection, but here the selection is at the kernel level. Again, in a class-specific learning step, the descriptors (appearance or shape or both) most suitable for a particular class are determined. For example, a category such as *car* is best described by shape alone, *leopard* by appearance alone, and *buddha* by a combination of the two.

Automatically ROIs detection

For training sets that are not constrained in pose or that have significant background clutter, to describe them with the PHOW and PHOG for the whole image (treating image classification as scene matching) is not sufficient. Instead it is necessary to “home in” on the object instance in order to learn its visual description. To this end we automatically learn a Region Of Interest (ROI) in each of the training images. These regions can be identified from the clutter by measuring similarity using the PHOW and PHOG but here defined over a ROI rather than over the entire image. The result is that "clean" visual exemplars [12] are obtained from the pose varying and cluttered training images.

Improving classification efficiency

We use random forest classifier to increase the speed of our system. The advantage of randomized trees is that they are much faster in training and testing than traditional classifiers (such as an SVM). They also enable different cues (such as appearance and shape) to be "effortlessly combined" [153]. These classifiers have been applied to object recognition but only for a relatively small number of classes. Here we increase the number of object categories by an order of magnitude (from 10 to 256). The research question is how to choose the node tests so that they are suited to spatial pyramid representations and matching. The novelty of our approach is that the classifier has the ability to choose the weight given to shape, appearance and the levels of the pyramids of each. This facilitates representations and classification suited to the class without weight optimization on a validation set.

1.5 Outline of the thesis

The structure of the thesis is as follows: In chapter 2 we review existing work in the field of image classification focusing on the different image representation models and most frequent used descriptors. Special attention is given to the bag-of-words representation using local image regions, which form the basis for this thesis work. Chapter 3 introduces the variety of datasets used in the thesis to assess the proposed method. In chapter 4 we propose the hybrid generative/discriminative approach for image classification. We study and discuss in depth how the different parameters affect the final classification performances. In chapter 5 we add spatial information to the hybrid model and discuss its advantages and disadvantages over previous works. In chapter 6 we introduce a new shape descriptor and study how different features can be used together for image classification. A faster classifier and a new algorithm to automatically detect the salient objects in images are introduced in chapter 7. Finally, in chapter 8 we draw a summary and discussion about this thesis and discuss future work.

The most used terminology and abbreviations are summarized in appendix A and a medical image application for breast parenchymal tissue classification is shown in appendix B.

Chapter 2

Literature review

In this chapter we will review the most recent and significant works in the literature on image classification. We first discuss the different ways to represent the images in section 2.1. Pioneering works on image classification used color, texture and shape features directly from the image in combination with supervised learning methods to classify images into several categories (e.g indoor, outdoor, sunset). The problem of image representation using low-level features has been studied in image and video retrieval for several years and we review them in section 2.1.1. Later works proposed to model the images by a semantic intermediate representation in order to reduce the gap between low-level and high-level image processing. The purpose of these methods is to match the image model with the perception we humans have (e.g. a street scene mainly contains road and buildings). These methods are reviewed in section 2.1.2. Then, in section 2.1.3 we pay special attention to the most recent representation methods which use local regions. We review the bag-of-words approaches and the region detectors used. These methods have been the most used over the past 5 *years and it has been shown to obtain very good performances when used for image classification. In the second part of this review (section 2.2) we discuss the different appearance and shape descriptors used for image classification.*

2.1 Image representation

In the literature many techniques have been used to represent the content of an image. Here we classify them into three main approaches: (i) those methods which directly extract low-level features from the images, (ii) those methods which use an intermediate semantic representation of the image, and (iii) those methods which model the image using local patches as intermediate representation. The philosophy of each approach as well as the main methods are described here below.

2.1.1 Low level image representation models

The problem of image classification is often approached by representing the images using low-level features (e.g. power spectrum, color histograms). This representation is then used to classify the images into a category (e.g. *street*, *dolphin*). These methods consider that images can directly be described by their low-level properties. For instance, a *forest* scene presents highly textured regions (trees), the presence of straight horizontal and vertical edges denotes an *urban* scene, a red color represents a *stop sing*, and blue color represents a *coast* or a *dolphin* image etc. Amongst these methods we can distinguish two approaches: (i) global representations where the low-level features are computed over the whole image, and (ii) local representations where the image is first partitioned into several blocks, and then features are extracted from each of these blocks. Figure 2.1a shows an example of a global model representation and figure 2.1b shows an example of a local model representation, both using low-level features. Theses two models are reviewed here below.

Global models

Vailaya *et al.* [143, 144, 145] consider the hierarchical classification of vacation images, and show that a global representation can successfully discriminate between many scenes types using a hierarchical structure. Using binary Bayesian classifiers, they attempt to capture the image category from global image features under the constraint that the test image belongs to one of the classes. At the highest hierarchical level, images are classified as *indoor* or *outdoor*; outdoor images are further classified as *city* or *landscape*; finally, a subset of landscape images is classified into *sunset*, *forest*, and *mountain* categories. Different qualitative measures, extracted from the whole image, are used at each level depending on the classification problem: indoor/outdoor (using spatial color moments); city/landscape (edge direction coherence vectors), and so on. The classification

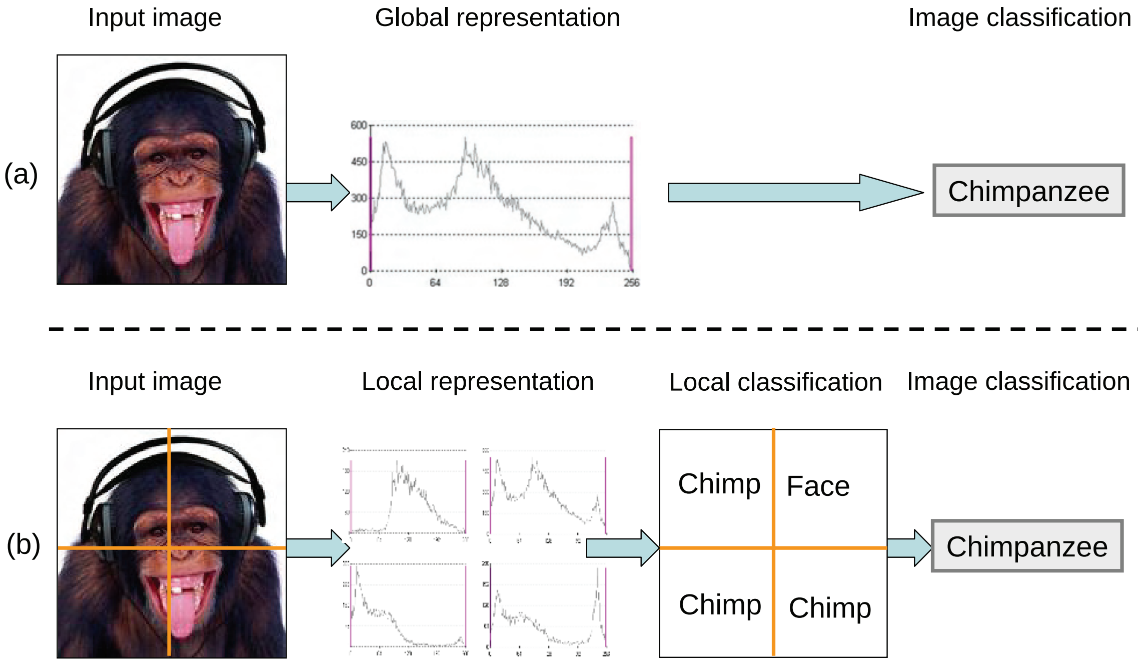


Figure 2.1: Example of global and local image representation. (a) Global image representation by using low-level features (e.g. a color histogram). This representation is the input of the classifier and a final category is given; (b) local image representation by using low-level features (e.g. a color histogram at each sub-block). Each subblock is independently classified obtaining a category for each one. These results are finally combined to obtain an image category.

problem is addressed by using Bayes decision theory. The proposal reports an excellent performance over a set of 6931 images.

Chang *et al.* [32] use a global image representation to produce a set of category labels with a certain belief for each image. They manually label each training image with a category and train k classifiers (one for each category) using Support Vector Machines (SVM). Each test image is classified by the k classifiers and assigned a confidence score for the category that each classifier is attempting to predict. As a result, a k-nary label-vector consisting of k-class membership is generated for each image. This approach is specially useful for Content Based Image Retrieval (CBIR) and Relevance Feedback (RF) systems. Other authors have followed this global approach, although they have taken other aspects into account. For example, Shen *et al.* [125] makes emphasis on the type of features that must be used. The authors argue that due to the complexity of visual content, a classification system can not be achieved by considering only a single type of feature such as color, texture and shape alone and proposed Combined Multi-Visual Features. It produces a low-dimensional feature vector useful for an effective classification. Their method is tested on image classification using three different classifiers: SVM, K-Nearest Neighbors (K-NN) and Gaussian Mixture Models (GMM). Other authors use global edges or orientation histograms [138].

Local models

Theses approaches are a direct extension of the global low-level approaches described above. The global approaches use low-level features extracted from the whole image, while the local ones first split the image into a set of subregions, which are further represented by their low-level properties. Each block is then classified as a certain category and finally the image is categorized from the individual classification of each block.

The origin of this approach can be found in 1997, when Szummer and Picard [134] proposed to independently classify image subsections to obtain a final result using a majority voting classifier. The goal of this work was to classify images as indoor or outdoor. The image is first partitioned into 16 sub-blocks from which Ohta-space color histograms and MSAR texture features are then extracted. K-NN classifiers are employed to classify each sub-block using the histogram intersection norm. Finally the whole image is classified using a majority voting scheme from the sub-block classification results. They demonstrated that performance is improved by computing features on sub-blocks, classifying these sub-blocks, and then combining theses results in a way reminiscent of stacking.

Similar results were also obtained by Paek and Chang [110]. Moreover, they developed a framework to combine multiple probabilistic classifiers in a belief network. They trained classifiers for *indoor*/*outdoor*, *sky*/*no sky* and *vegetation*/*no vegetation* as secondary cues for the *indoor*/*outdoor* problem. The classification results of each one are then feeded into a belief network to take the integrated decision.

The proposal of Serrano *et al.* [123] in 2004 shares this same philosophy, but using SVM for a reduction in feature dimensionality without compromising classification accuracy. Color and texture features are also extracted from image sub-blocks and separately classified. Thus indoor/outdoor labels are obtained for different regions of the scene. The advantage of using SVM instead of K-NN classifier is that the sub-block beliefs can be combined numerically rather than by majority voting, which minimizes the impact of sub-blocks with ambiguous labelling.

Discussion

The main advantage of these methods is that they provide a very simple image representation. The main drawback of representing the images by low level features is that if images have a notable background clutter or there is lot of intra-class variability, this representation is not enough to discriminate amongst different categories. Some authors which use this representation argue that the images they use have low intra-class variation and can be easily separated by using low-level features. Moreover these methods have been used to classify amongst a few number of image categories (from 2 to 5).

2.1.2 Semantic image representation

The meaning of the semantic of an image is not unique, so we can find in the literature different semantic image representations. We can distinguish between two semantic meanings: (i) those methods which globally represent the semantic properties related to the image structure such as ruggedness, expansiveness, etc., and (ii) those methods which identify the "local semantic objects" (e.g. *sky*, *building*, *car*) that appear in the image and use this object occurrences to classify the image as a category. We will refer to these methods as global and local models respectively. Both are following reviewed.

Global models

These works make use of a semantic description by using the statistical properties of the image. They introduce an intermediate semantic level related to global configurations and image structure. Therefore the image is described by visual properties, which are shared by images of a same category.

Oliva and Torralba [102, 103, 140] proposed a computational model for the recognition of real world scenes (4 natural scenes and 4 man-made scenes). The procedure is

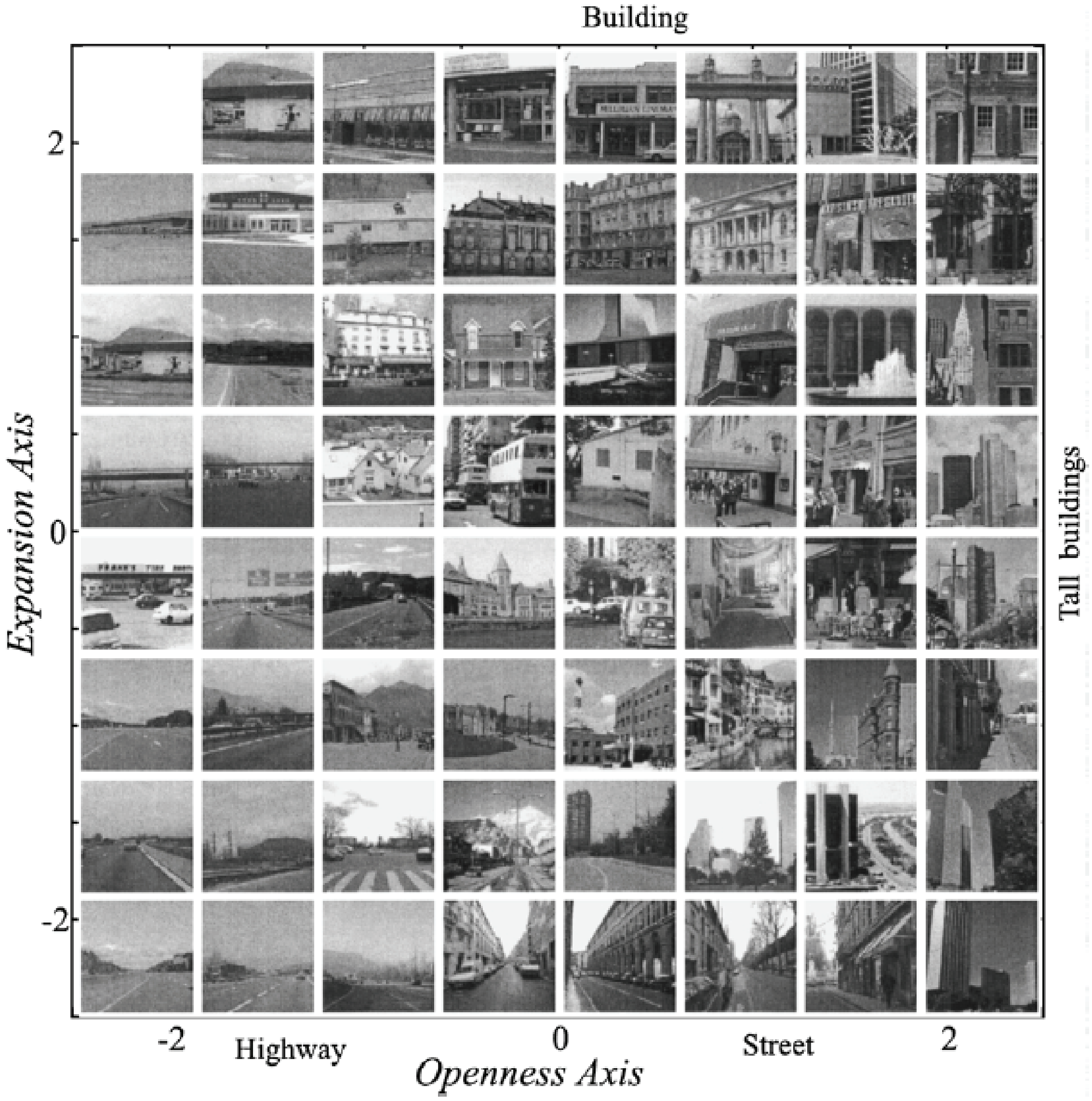


Figure 2.2: Organization of man-made environments according to the semantic degrees of openness and expansion [102].

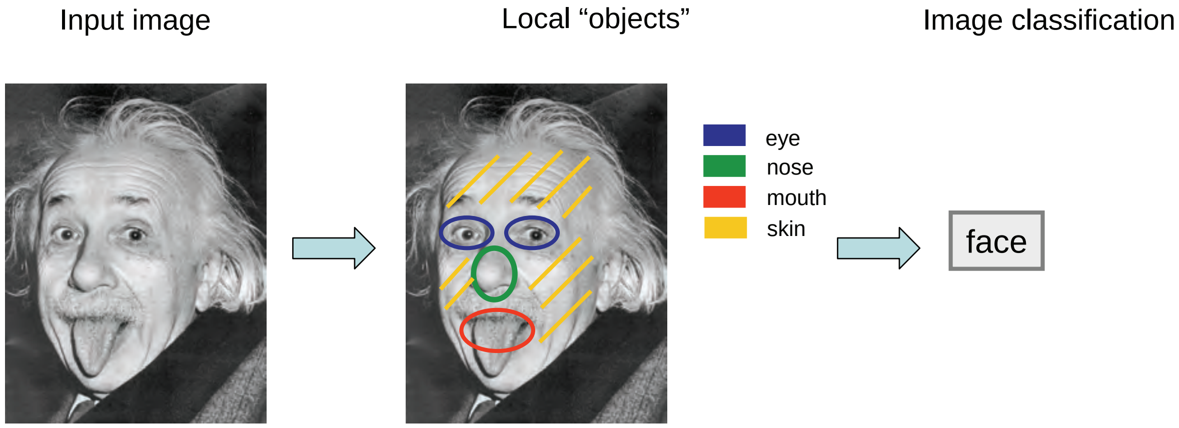


Figure 2.3: Overview of a semantic representation using a local model.

based on a very low dimensional representation of the scene, that they refer to as the Spatial Envelope. It consists of five perceptual qualities: naturalness (vs man-made), openness (presence of a horizon line), roughness (fractal complexity), expansion (perspective in man-made scenes), and ruggedness (deviation from the horizon in natural scenes). Each feature corresponds to a dimension in the spatial envelope space, and together represent the dominant spatial structure of a scene. Then, they show that these dimensions may be reliably estimated using spectral and coarsely localized information. The model generates a multidimensional space in which scenes sharing membership in semantic categories are projected close together (see figure 2.2 for an example). Therein it is possible to assign a specific interpretation to each dimension: along the openness dimension, the image refers to an open or a closed environment, etc..

Local models

Local semantic content of the images may be used as an intermediate representation for image classification allowing to deal with the gap between low- and high-level features. For example the presence of *streets*, *cars* and *buildings* denotes a urban scene, or the presence of *eyes*, *nose*, *mouth*, denotes a *face* image. This is illustrated in figure 2.3

These methods are mainly based on first localizing the different image regions. Then local classifiers are used to label the regions as belonging to an object (e.g. *sky*, *people*, *wheel*). Some times it is also introduced some spatial relationships between objects in the images (e.g. the *sky* is above a *mountain* or the *eyes* are above the *nose*) [48, 88]. Finally, using this local information, the global image is classified. Different ways to carry out the image classification using this strategy have been proposed recently.

Mojsilovic *et al.* [96] first segment the image using color and texture information to find the semantic indicators (e.g. *skin*, *sky*, *water*). Then, these objects are used to identify the semantic categories (e.g. *people*, *cars*, *landscapes*). A similar approach is proposed in [49]. Fan *et al.* [39] introduce a set of functions (learned from the labelled image regions) each one used to detect a specific type of object. The class distribution of the image category is approximated using finite mixture models of its objects. A new test image is classified as the category with maximum posterior probability. In addition, a large number of unlabelled samples are integrated with a limited number of labelled samples to achieve more effective classifier training and knowledge discovery. Interactions between the objects in terms of their spatial relationships is introduced by Aksoy *et al.* [1]. Initially an image segmentation is performed using a classical split-and-merge algorithm. Then, the technique automatically learns representative region groups which discriminate different images and builds visual grammar models.

Barnard *et al.* [7] present an approach for modelling multi-modal data sets, focusing on the specific case of segmented images with associated text. They consider in detail predicting words associated with whole images (auto-annotation) and corresponding to particular image regions (region naming). Auto-annotation might help organize and access large collections of images. They develop a number of models for the joint distribution of image regions and words, and study multi-model and correspondence extensions of Hofmann's hierarchical clustering/aspect mode, a translation model adapted from statistical machine translation, and a multi-modal extension to mixture of latent Dirichlet allocation. A similar work was presented in [38]. Singhal *et al.* [127] proposed a holistic approach to determining scene content (objects) based on a set of individual material detection algorithms, as well as probabilistic spatial context models.

In contrast to previous approaches that first use a segmentation step, Vogel and Schiele [150] use a spatial grid layout which splits the image into regular subregions. The technique uses both color and texture to perform landscape image classification and retrieval based on a two-stage system. First, the image is partitioned into 10×10 subregions, and each one is classified using K-NN or SVM. An image is then represented by a so-called

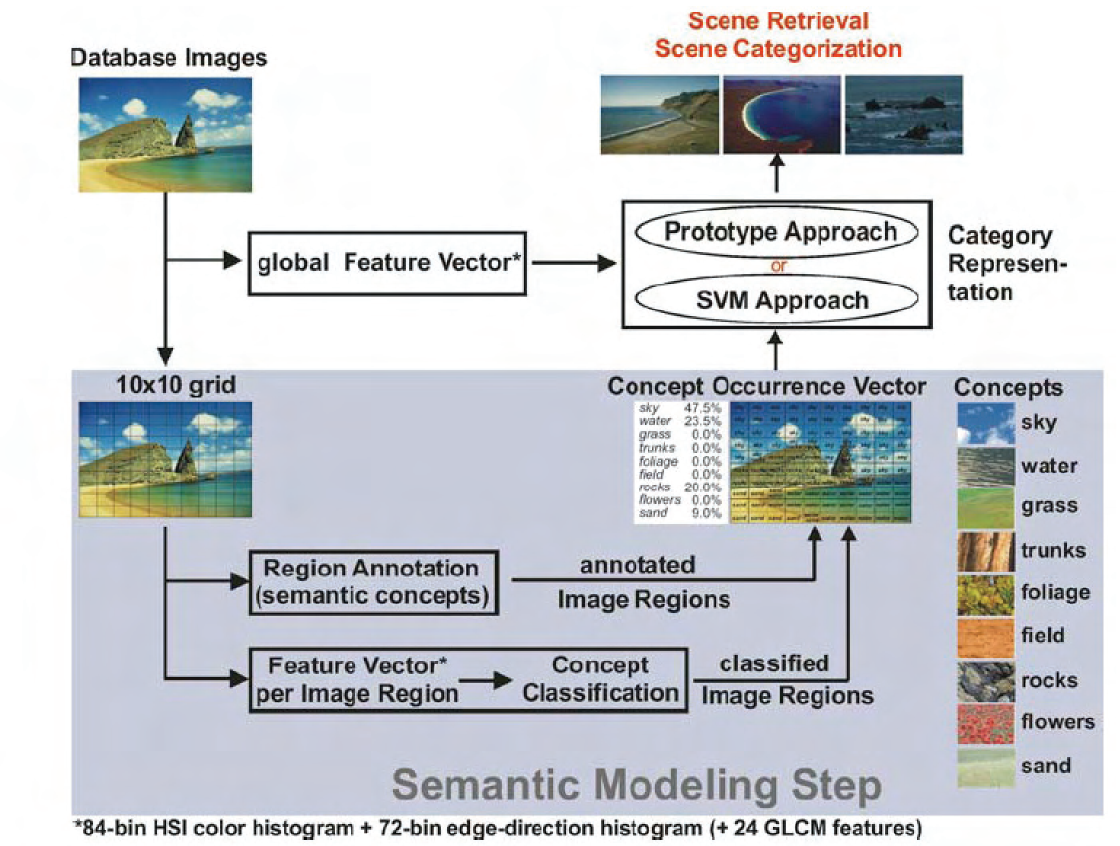


Figure 2.4: Overview of the approach proposed by Vogel and Schiele [150].

concept occurrence vector (COV), which measures the frequency of different objects in a particular image. Given this image representation, a prototypical representation for each scene category can be learnt. Image classification is carried out by using the prototypical representation itself or Multi-SVM approaches. This approach is illustrated in figure 2.4

Sudderth *et al.* [133] describe a hierarchical probabilistic model for the detection and recognition of objets in cluttered, natural scenes. The model is based on a set of parts which describe the expected appearance and position, in an object centered coordinate frame, of features detected by a low-level interest operator. Each object category then has its own distribution over these parts, which are shared between objects. Object appearance information is shared between all scenes in which that object is found.

An hybrid approach is proposed by Luo *et al.* [85]: low-level and semantic features are integrated into a general-purpose knowledge framework that employs a Bayesian Network. The efficacy of this framework is demonstrated via three applications involving semantic understanding of pictorial images: (i) detection of the main photographic subjects in an image [86], (ii) selecting the most appealing image in an event, and (iii) classifying images into *indoor* or *outdoor* scenes. The performance is quantitatively evaluated using only low-level features (Ohta color space histograms and MSAR texture features as in [134]), and incorporating "semantic features" (*sky* and *grass*). They demonstrate that the classification performance can be significantly improved when local semantic features are employed in the classification process.

Discussion

The main advantage of theses methods is that they use human meanings to first classify the objects and then the image. It gives a more powerful, discriminative representation and actually these methods have been applied to classify images into a big number of categories than the low-level methods. The main drawback is that most of them are first based on segmenting the image and this can cause some problems when working with complex images. If the segmentation method is not accurate it can merge some parts of the objects causing a bad image description. Other problems can come when the object discovery fails, because the further image classification is based on the object occurrences. So a wrong object classification probably implies an erroneous final image classification.

Furthermore, Thorpe *et al.* [137] found that humans are able to categorize complex natural images containing animals or vehicles very quickly. Fei-Fei *et al.* [42] later showed that little or no attention is needed for such rapid natural image categorization. Both of theses studies posed a serious challenge to the currently accepted view that to understand the context of a complex scene, one needs first to recognize the objects and then in turn recognize the category of the image [141].

2.1.3 Local patches representation

In this case images are represented by hundreds of local patches. They use a region detector to find a set of interesting parts of the image and then represent them by some kind of descriptor. In recognition a matching between the region descriptors of the new image and those in the database is computed. The new image is classified if sufficient matches occur. These methods can be extended by applying geometric constraints. The use of local descriptors has become popular for object detection and recognition. For example Fergus *et al.* [45] model object classes as probabilistic constellations of parts. The appearance of each part, as well as pair-wise relations between parts, are modelled using Gaussian distributions. The model can be nicely visualized as a collection parts connected by springs, so that parts can move with respect to each other. The learning algorithm automatically looks for a configuration of detected image regions consistent over the training data. Recognition proceeds by first detecting potential part locations in an image, and then comparing hypotheses as to whether observed features are generated by the category model or by the background model. A Bayesian extension of the constellation model of Fergus *et al.* [45], capable of learning from a small number (3-5) of training images, was presented by Fei-Fei *et al.* [40]. This work shows that knowledge about other object classes, here in the form of a prior, can help in learning new object class models. The work of Zhang *et al.* [157] is another example of the successfulness of the bag-of-words representation.

Bag-of-Words model

In the last years we can find in the literature on image classification, an increasing number of proposals which make use of the bag-of-words model representing the images using histograms of quantized appearances of local patches [35, 108, 157, 158]. All these proposals represent the image content by local descriptors, for example visual words [75, 79, 115, 147].

The bag-of-words methodology (sometimes also called bag-of-features or bag-of-visterns) was first proposed for text document analysis and further adapted for computer vision applications [79, 129]. The models are applied to images by using a *visual* analogue of a *word*, formed by vector quantizing visual features (color, texture, etc.) like region descriptors. Recent works have shown that local features represented by bags-of-words are suitable for image classification showing impressive levels of performance [41, 76, 116]. Constructing the bag-of-words from the images involves the following steps: (i) Automatically detect regions/points of interest (local patches), (ii) compute local descriptors over these regions/points, (iii) quantize the descriptors into words to form the visual vocabulary, (iv) find the occurrences in the image of each specific word in the vocabulary in order to build the bag-of-words (histogram of words). Figure 2.5 schematically describes

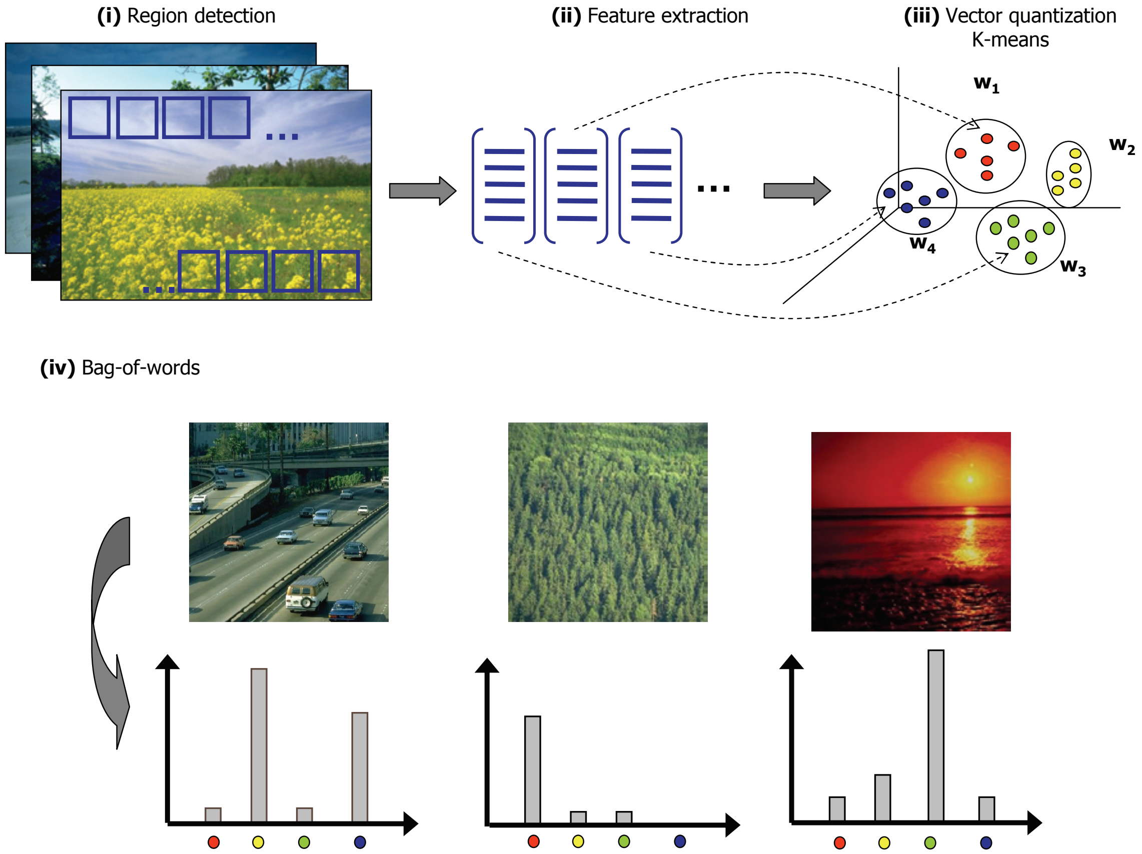


Figure 2.5: Four steps to compute the bag-of-words when working with images. (i-iii) obtain the visual vocabulary by vector quantizing the feature vectors, and (iv) compute the image histograms – bag-of-words – for images according the obtained vocabulary.

the four steps involved in the definition of the bag-of-words model.

The advantage of this method is its simplicity and the relatively small amount of supervision required. Labelling training data only requires indicating the image category. Moreover these methods have been used to classify images into a big number of categories (up to 100).

First works using the "bag-of-words" representation can be found in the literature related to texture classification. The goal of these works is to recognize textures captured from different camera viewpoints, and under varying illumination. Leung and Malik [79] quantized responses of a filter bank applied densely over an entire image. These quantizations of appearance descriptors are called "Textons" and textures are represented by distributions of textons. Varma and Zisserman [147] modified this approach by quantizing small image patches rather than filter responses. Lazebnik *et al.* [75] address texture classification using quantized affine covariant regions.

Recently, these representations have been extended for object recognition and scene classification. Perronnin *et al.* [111] defined a universal vocabulary, which describes the content of all the considered images, and class visual vocabularies which are obtained through the adaptation of the universal vocabulary using class-specific data. While previous approaches characterize an image with a single histogram, here an image is represented by a set of histograms, one per class. Each histogram describes whether an image is more suitably modelled by the universal vocabulary or the corresponding adapted vocabulary. They represent a vocabulary of visual words by means of a GMM where $\lambda = w_i, \mu_i, \Sigma_i, i = 1..N$. λ denotes the set of parameters of a GMM, w_i, μ_i and Σ_i denote respectively the weight, mean vector and covariance matrix of Gaussian i and N denotes the number of Gaussians. Each Gaussian represents a word of the visual vocabulary. The Universal vocabulary is trained using maximum likelihood estimation (MLE) and the class vocabularies are adapted using the maximum a posteriori (MAP) criterion. They successfully test the method classifying images like *sunset*, *underwater*, *cars*, *bikes*.

Some Bayesian models used for text document classification, such as Latent Dirichlet Analysis (LDA) and probabilistic Latent Semantic Analysis (pLSA), work over the bag-of-words model and have been adapted and used to model image categories. Li and Perona [41] independently proposed two variations of LDA firstly proposed by Blei *et al.* [16, 135] which was designed to represent and learn document models. In this framework, local regions are first clustered into different intermediate themes, and then into categories. Probability distributions of the local regions as well as the intermediate themes are both learnt in an automatic way, bypassing any human annotation. No supervision is needed apart from a single category label to the training image.

Quelhas *et al.* [116] provided an approach by bag-of-words to model visual scenes in image collections, based on local invariant features and pLSA. pLSA is a generative model from the statistical text literature [65]. In text analysis this is used to discover topics in a document using the bag-of-words document representation. In this case, there are “images” as “documents” and they discover “topics” as “object categories” (e.g. *grass*, *houses*, *bikes*, *planes*), so that an image containing instances of several objects is modelled as a mixture of topics. pLSA, an unsupervised probabilistic model for collections of discrete data, has the dual ability to generate a robust, low-dimensional scene representation, and to automatically capture meaningful scene aspects. They successfully used the first property for scene classification, and have exploited the second one to design two new algorithms: one for aspect-based image ranking, and another for context-sensitive image segmentation. pLSA was also used in [128] for object recognition.

Variations of latent space models have recently been applied to the problem of modelling annotated images [7].

Habitual bag-of-words techniques, as the described above, do not take the spatial information into account. However, in complex natural images, image classification systems can be further improved by using contextual knowledge like common spatial relationships between neighboring local objects [88] or the absolute position of objects in certain scenes [150]. While the above methods have shown to be effective, their neglect of spatial structure ignores valuable information which could be useful to achieve better results for image classification.

Lazebnik *et al.* [76] proposed a method which is based on spatial pyramid matching of Grauman and Darrell [57]. Pyramid matching works by placing a sequence of increasingly coarser grids over the feature space (in this case over the image) and taking a weighted sum of the number of matches that occur at each level of resolution (L). At any fixed resolution, two points are said to match if they fall into the same bin of the grid; matches found at finer resolutions are weighted more highly than matches found at coarser resolutions (α_l represents the weight at level l). The resulting spatial pyramid is an extension of the bag-of-words image representation, it reduces to a standard bag-of-words when $L = 0$ (see figure 2.6). Multi-class classification is done with SVM. This method achieves high accuracy on a large database of 15 natural scene categories and on the well know Caltech-101 dataset. Fergus *et al.* [44] develop two new models, ABS-pLSA and TSI-pLSA, which extend pLSA to include absolute position and spatial information in a translation and scale invariant manner respectively.

Support regions

Salient intensity regions in images are the most used support patches for image classification. In this thesis we have chosen to use the similarity-invariant regions and affine-

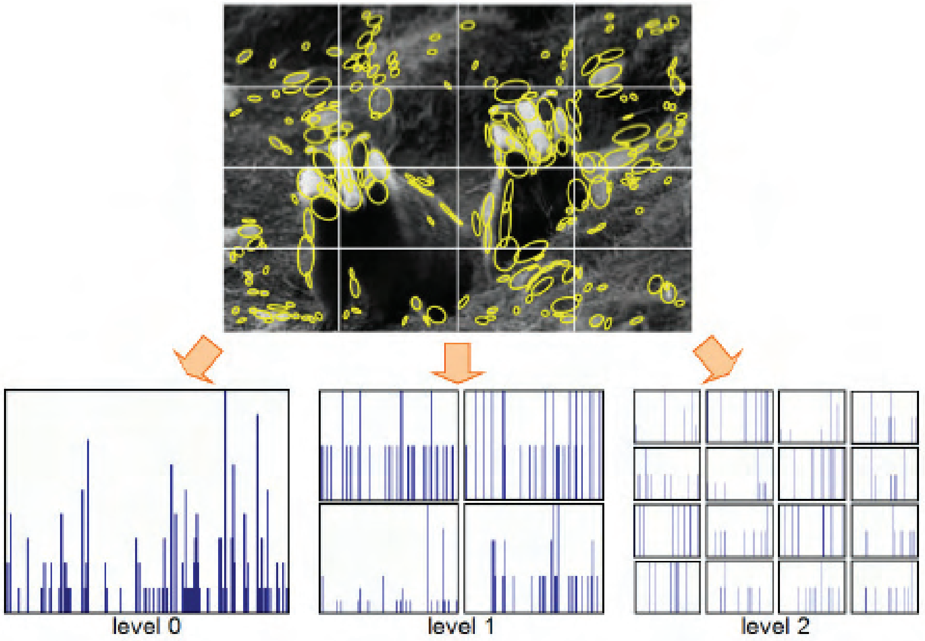


Figure 2.6: The Pyramid matching method of Lazebnik *et al.* [76]. Set of histograms computed over a multi-level pyramid decomposition of the image.

covariant regions. However other support regions can be found in the literature [68, 84, 91, 142]. A comprehensive review of affine covariant region detectors, and a comparison of their performance appeared in Mikolajczyk *et al.* [94].

Similarity-invariant regions. Early work includes interest point detection ([99], [60]), which was based on detecting regions around corners as local structures with high information content. This methods were based on first detecting interest points in the images (e.g. corners) and then open a patch around these points. The patch was normally a circular or a squared patch with a fixed size. Blostein and Ahuja [17] were the first to introduce a multiscale region detector based on maxima of the Laplacian. Lindeberg and Garding [83] have extended this detector in the framework of automatic scale selection. Other methods use a regular grid of densely sampled patches over the entire image.

Affine-covariant regions. Affine-covariant region detectors were investigated by Mikolajczyk and Schmid [92] and by Schaffalitzky and Zisserman [120]. They are designed to find corners within images, adjusting the shape of the region to find an optimal size. This is done in two steps (i) use the Harris interest point detector over a range of image

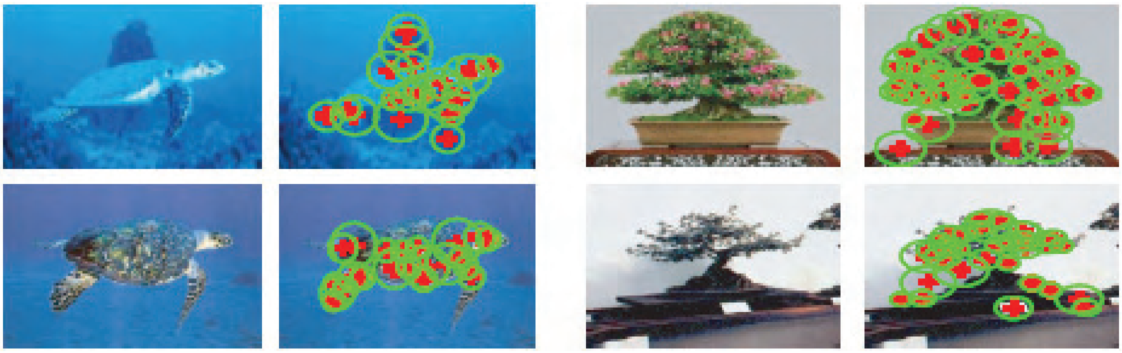


Figure 2.7: Images with and without the feature detector [68] superimposed.

scales and keep those points which are local maxima over scale evaluated by a Laplacian of Gaussian operator; (ii) an iterative procedure is then used to adapt the shape of the circular point neighborhood of each interest point to the skew normalized frame, where second moment matrix of image gradients is isotropic. This is done by varying the radius of a scale-normalized Laplacian filter and measuring its response when applied to the image at the interest point. The response will have a peak at a certain radius, which is taken to be the characteristic shape. The original idea of iterative shape adaptation using the second moment matrix is due to Lindeberg and Garding [51].

Figure 2.7 shows some images with the feature detector imposed. Affine-covariant regions are appropriated when working with textured images because they allow to find representative regions even when scale and view-point variations. Figure 2.8b shows the results when applying these feature detectors in natural scenes. As is shown, the result is a sparse representation of the image and we can see that the output is very poor and images are not very well represented. To solve this problem a regular grid -dense representation- (figure 2.8c) over the image is often used. In this way we can have information of the whole image and not only from those parts where objects or interest points are detected. Similarity-invariance is often sufficient for scene classification because the training data covers other viewpoint changes. Regular grids have been recently used for pedestrian detection [36] and in [41] it has been demonstrated better performance than when using sparse regions.

Figure 2.8: Images with the feature detector superimposed: (a) original image; (b) Harris affine is superimposed in images representing natural scene; (c) a regular grid is used to represent regions in natural images.

2.2 Image descriptors

2.2.1 Appearance information

Detection of local image regions is only the first part of the feature extraction process; the second part is the computation of descriptors to characterize the appearance of these regions. A goal descriptor should be distinctive, so as to provide strong consistency constraints for image matching yet robust to illumination changes and other appearance variations. We now review a variety of the most recent descriptors used for image classification which are based on appearance information.

First Koenderink and van Doorm [71] proposed a method which describes a region using local derivatives up to a given order. This is called “local jet” or differential invariants. A set of filter are also used to describe intensity regions: Gabor filters [87], steerable filters [50] or more complex filters presented in [121]. Van Gool *et al.* [146] presented the moment invariants, a collection of affine and photometric invariants for planar regions that describe the shape and intensity distribution. Another technique using histograms on the intensity values is the spin image [67].

Most of the recently proposed image classification methods used the Scale Invariant Feature Transform (SIFT) descriptor proposed by Lowe [84]. This descriptor takes each region, finds its gradients and then normalizes for orientation by finding the dominant orientation rotating the region so as to make it axis aligned. Then 8-bin orientation histograms are formed of the gradients in each cell of a 4×4 spatial grid overlaid on the region. Each region is described by a $4 \times 4 \times 8 = 128$ dimensional vector (figure 2.9 shows an overview of this method). The idea is that the loose grid gives a little bit of slop to accommodate minor translation and scale offsets due to inexact feature detection while the gradient based representation makes it less sensitive to illumination changes. It has been demonstrated that these features achieve a higher performance for object classification [94].

Some extensions of the original SIFT features exist. PCA-SIFT [70] and the Gradient location and orientation histogram (Gloh) [94] from which change the location grid and use PCA for dimensionality reduction. A different matching scheme called SURF (Speeded Up Robust Features) was presented by Bay *et al.* [9]. The standard version of SURF is faster than SIFT and proved to be more robust against different image transformations than SIFT. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images.

Color is an important component of the natural image categories [119]. However, outdoor scenes are especially complex to deal with in terms of the lighting conditions and the fact that color-based features suffer from the problem of color constancy [72]. Buluswar and Draper [24] provided a survey detailing analysis and causes of color variation due to illumination effects in outdoor images. Color being undoubtedly one of the most interesting characteristics of the natural world, can be computationally treated in many different ways. In many cases the basic RGB components may provide a valuable information about the environment. However, the perceptual models, such as CIE or HSI, are more intuitive and therefore enable the extraction of characteristics according to the

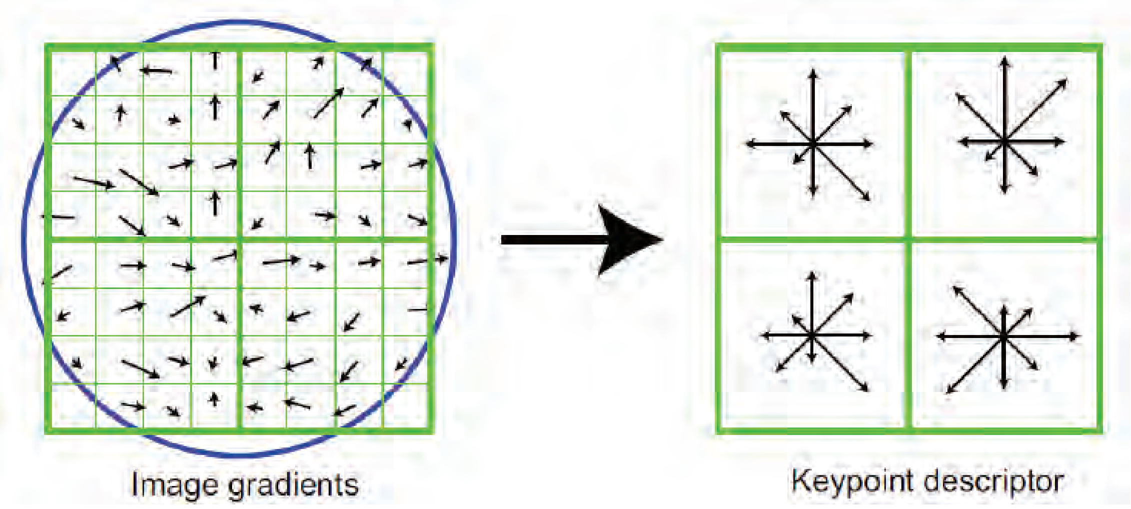


Figure 2.9: The SIFT descriptor of Lowe [84]. On the left are the gradients of an image patch. The blue circle indicates the Gaussian center-weighting. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. A 2×2 descriptor array computed from an 8x8 set of samples is shown here.

model of human perception. Invariance and discriminative power of the color invariants is experimentally investigated in [54], showing the invariants to be successful in discounting shadow, illumination, highlights, and noise.

Some authors, like Ohta [101], have proposed their own color space. Celenk [30] proposed operating with the CIE(L*, a*,b*) uniform color coordinate system L*, H^o and C* (Luminance, Hue and Chroma). This color space defines approximately a space having uniform characteristics. Campbell *et al.* [28] also proposed a set of color parameters in order to work with outdoor scenes. Similarly, Mori *et al.* [98] proposed the use of the r-b model (where r and b denote normalized red and blue components respectively) in order to solve the problems of hue shift, due to outdoor conditions and shadows. We can find in the literature some models based on the Hue, Saturation and Intensity (HSI) for example, the model described by Smith [130] or the alternative proposed by Tenenbaum [136]. Yagi *et al.* [156] calculates the hue and intensity based on Smith work, and proposed a different way to obtain the saturation. The most typical way to calculate the components HSI is described in [55].

In 2002, Buluswar and Draper [25] developed models for illumination and surface reflectance to be used in outdoor color vision, and in particular to predict the colour of surfaces under various outdoor conditions. Moreover, demonstrated the disadvantages of using the CIE model for predicting colour in outdoor images. More recently, in 2004, Berwick and Lee [13] presented a framework of logarithmic chromacities for the interpretation of the image colour change due to illumination pose and colour. Some approaches of using colour with SIFT features have also been proposed recently in [53, 152].

2.2.2 Shape information

It has been demonstrated that shape represented by the edges is a very good cue for object recognition [108, 126]. Thus, we would like to give a fast review of some of the techniques used to detect and describe the shape information.

Basic techniques for edge detection include the Sobel/Prewitt edge detector [56]. Canny [29] developed an improved approach to find edges. Since then various others made slight improvements in various directions [37, 124]. Recently Martin *et al.* [89] have proposed the Berkeley natural boundary detector which has reported excellent results.

Several methods to describe and compare edges have been proposed. One popular method is Chamfer Matching which was introduced by Barrow *et al.* [8] and extended to hierarchical matching by Borgefors [19]. The algorithm matched edges by minimizing a generalized distance between them. The matching is performed in a series of images depicting the same scene, but in different resolution (e.g. in a resolution pyramid). Olson and Huttenlocher [107] used the Hausdorff-Distance to compare edges. They also showed how such a technique improves if edge orientation is taken into account. Another method for describing edges is the shape context descriptor of Belongie *et al.* [10]. There each point on an edge is characterized by the histogram of the log-polar coordinates (related to that specific point) of all other points (in a certain radius). Rothwell *et al.* [118] presented a method where edges are described between bitangent points. This results in a projectively invariant description for near planar curves. This method is an extension of the affine invariant representation of Lamdan *et al.* [73].

One of the first approaches for detecting objects of a category which was useful for real world images and based on shape was introduced by Gavrila and Philomin [52]. Sets of significant contour examples were used and then the whole contours were applied to test images. Berg *et al.* [12] perform generic object recognition on the basis of deformable shape matching using a new correspondence finding algorithm. Their algorithm is formulated as an integer quadratic program, where the cost function is a combination of geometric blur descriptors and geometric distortion between feature points. The recognition procedure is incorporated in a nearest neighbor framework. Dalal and Triggs [36] use shape information in the form of grids of Histograms of Oriented Gradients (HOG). Studying influences of the binning of scale, orientation and position they yield excellent categorization by a SVM- based classifier. Leibe *et al.* [77] include shape information to detect pedestrians. They use a verification step that uses chamfer matching of a representation of the whole object contour. Recently, Shotton *et al.* [126] and Opelt *et al.* [108] presented a method based on local boundary fragments.

Chapter 3

Datasets

We will evaluate our classification algorithms on different datasets recently used in the literature. We can divide theses datasets in three groups: (i) Oliva and Torralba [102], Vogel and Schiele [149], Fei-Fei and Perona [41], Lazebnik et al. [76] are datasets containing natural scene images; (ii) Caltech 101 *[81] and Caltech-*256 *[59] are datasets with objects images, while (iii) TrecVid and Pretty woman datasets contain video images. In this chapter we will describe these datasets in more detail.*

3.1 Scene classification

3.1.1 Vogel & Schiele (VS) dataset

Vogel and Schiele [149] dataset (called as **VS** in this work): includes 702 natural scenes consisting of 6 categories: 144 *coasts*, 103 *forests*, 179 *mountains*, 131 *open country*, 111 *river* and 34 *sky/clouds*. Figure 3.1 shows some images from this dataset. The size of the images is 720×480 (landscape format) or 480×720 (portrait format). Every scene category is characterised by a high degree of diversity and potential ambiguities since it depends strongly on the subjective perception of the viewer. For example river and forest are considered as two kind of different scenes. However most of the river images also contain a forest and can easily be confused. There is a low inter-class variability and a high intra-class variability making the scene classification problem a bit more difficult when working with this dataset.

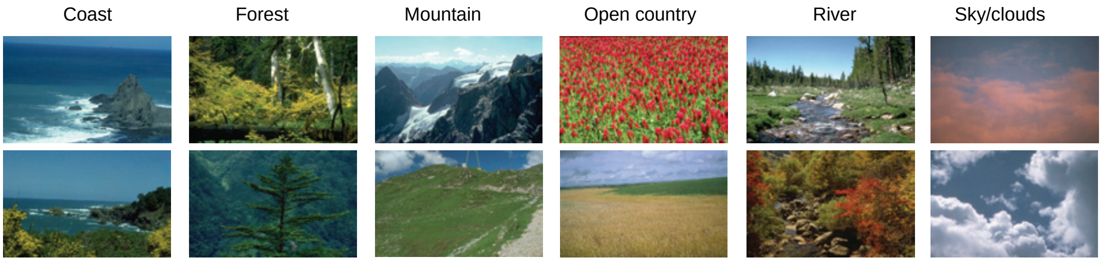


Figure 3.1: Some images from VS dataset.

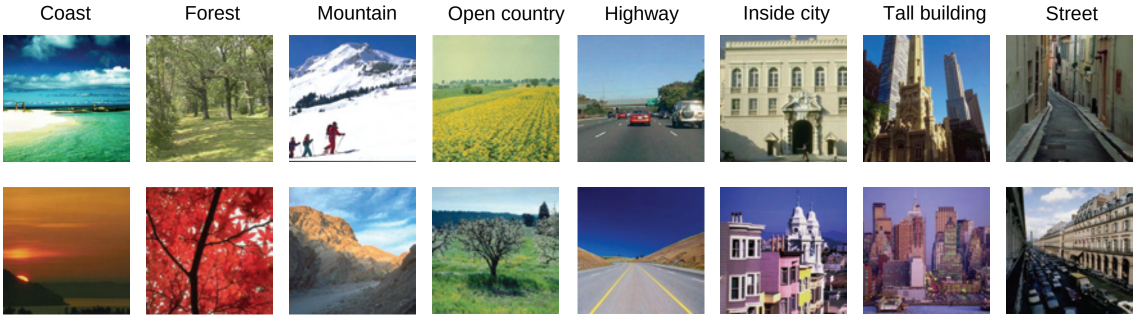


Figure 3.2: Some images from OT dataset.

3.1.2 Oliva & Torralba (OT) dataset

Oliva and Torralba [102] dataset (called as **OT** in this work) includes 2688 images classified as 8 categories: 360 *coasts*, 328 *forest*, 374 *mountain*, 410 *open country*, 260 *highway*, 308 *inside of cities*, 356 *tall buildings* and 292 *streets*. Figure 3.2 shows some images from this dataset. Note that now river and forest scenes are all considered as forest, moreover there is not an specific sky scene since almost all of the images contain the sky object. These annotations make a higher inter-class variability. Most of the scenes present a large intra-class variability. The average size of each image is 250×250 pixels.

3.1.3 Fei-Fei & Perona (FP) dataset

Fei-Fei and Perona [41] dataset (referred as **FP** in this work): contains 13 categories and is only available in greyscale. This dataset consists of the 2688 images (8 categories) of the OT dataset plus: 241 *suburb residence*, 174 *bedroom*, 151 *kitchen*, 289 *living room* and 216 *office*. Figure 3.3a shows some images from this dataset. Note that the presence

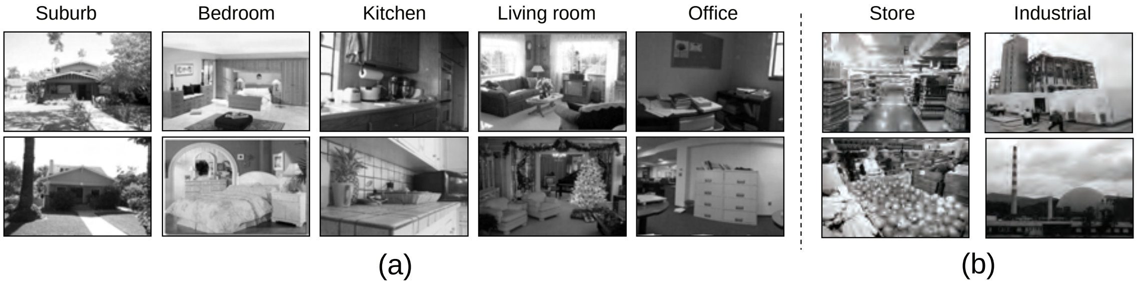


Figure 3.3: Some images from: (a) FP and LSP dataset and (b) LSP dataset.

of man-made images (e.g. bedroom, living room etc.) allows a low inter-class variability since these images have a similar structure. The average size of each image is approximately 250×300 pixels.

3.1.4 Lazebnik, Schmid & Ponce (LSP) dataset

Lazebnik *et al.* [76] dataset (referred as **LSP** in this work) contains 15 categories and, as with FP, is only available in greyscale. This dataset consists of the 13 categories of the FP dataset plus: 315 *store* and 311 *industrial*. Figure 3.3 shows some images from this dataset. The average size of each image is approximately 250×300 pixels.

3.2 Object classification

3.2.1 Caltech-101 dataset

The Caltech-101 dataset (collected by Fei-Fei *et al.* [81]) consists of images from 101 object categories. This database contains from 40 to 800 images per category however most categories have about 50 images. Most images are medium resolution, about 300×300 pixels. The significance of this database is its large inter-class variability. Moreover most images have little or no clutter, objects tend to be cantered in each image and most objects are presented in a stereotypical pose. So, as was noted by Lazebnik *et al.*[76], Caltech-101 is essentially a scene classification dataset as the objects are well aligned within each class (i.e. rotated, scaled and centred) with little clutter. An image from each category is shown in figure 3.4.

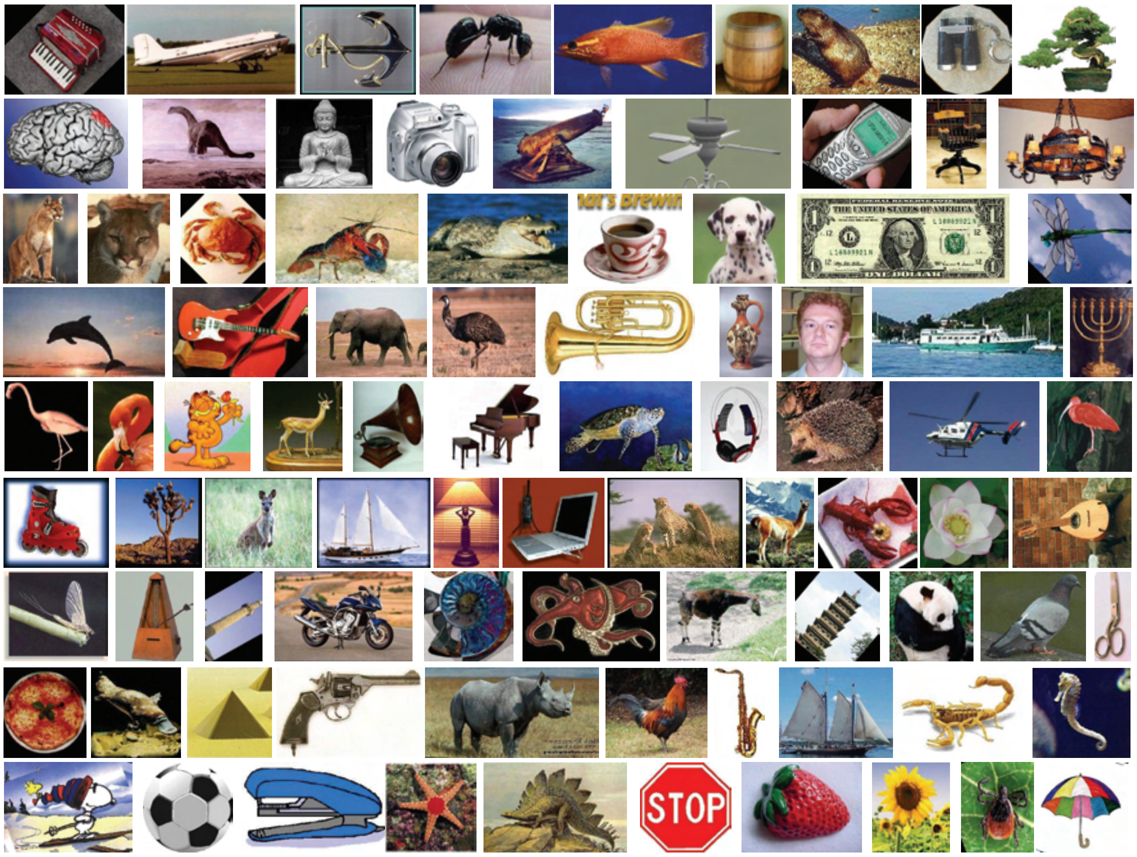


Figure 3.4: Some images corresponding to scenes/objects from Caltech-101 dataset.

3.2.2 Caltech-256 dataset

This data set (collected by Griffin *et al.* [59]) consists of images from 256 object categories and is an extension of Caltech-101. It contains from 80 to 827 images per category. The total number of images is 30608. The significance of this database is its large inter-class variability, as well as a larger intra-class variability than in Caltech-101. Moreover there is no alignment amongst the object categories. Fig. 3.5 shows 200 images from this dataset. For comparison to other authors, if otherwise stated, results will be provided for the first 250 categories.

3.3 Video data

We also evaluated our system using video image data. Concretely we used videos provided by TRECVid and *Pretty woman* movie.

3.3.1 TRECVid

We use the annotated training data from TRECVID 2006[1], which consists of 80 hours of video sequences. Video shots are annotated into 39 semantic categories which occur frequently in the database: (1) *sports*, (2) *entertainment*, (3) *weather*, (4) *court*, (5) *office*, (6) *meeting*, (7) *studio*, (8) *outdoor*, (9) *building*, (10) *desert*, (11) *vegetation*, (12) *mountain*, (13) *road*, (14) *sky*, (15) *snow*, (16) *urban*, (17) *waterscape-waterfront*, (18) *crowd*, (19) *face*, (20) *person*, (21) *government leader*, (22) *corporate leader*, (23) *police security*, (24) *military*, (25) *prisoner*, (26) *animal*, (27) *computer-TV-screen*, (28) *flag US*, (29) *airplane*, (30) *car*, (31) *bus*, (32) *truck*, (33) *boat-ship*, (34) *walking-running*, (35) *people marching*, (36) *explosion-fire*, (37) *natural disaster*, (38) *maps*, (39) *charts*.

TRECVID also provides keyframes for each shot. There are a total of 43907 keyframes. Some examples are shown in Fig. 3.6. Note the difficulty of these images, for example the mountain scenes are not as clear as the mountain scenes from OT dataset in figure 3.2. Only the keyframes are used here for learning and video shot retrieval.

The test data provided for TRECVID 2006 is only used for retrieval experiments. This data consists on about 100K keyframes from 160 hours of video sequences.

3.3.2 Pretty woman

Key frames from the movie *Pretty Woman* are used as test images to evaluate the scene classification system. The objective is to classify these images into a certain kind of scene. Note that this data is more real than previous datasets, there is people and more objects around, so that theses images will be more difficult to classify into one of the categories from previous datasets. We used every hundredth frame from the movie to form the testing set so that we have 1721 images. Figure 3.7 shows some key frames from this movie.

[1] http://www-nlpir.nist.gov/projects/trecvid/

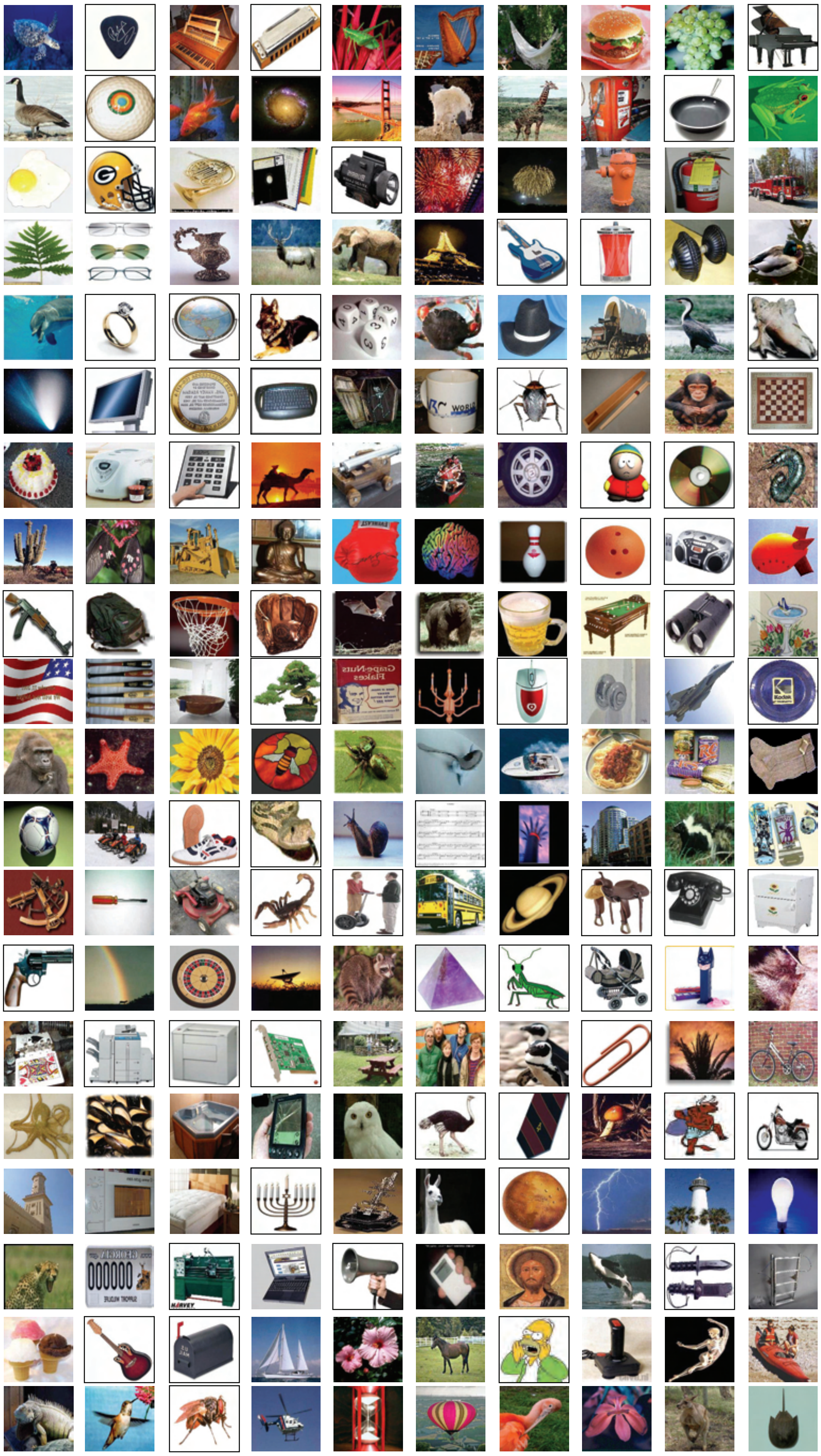


Figure 3.5: About 200 images corresponding to scenes/objects from Caltech-256 dataset.

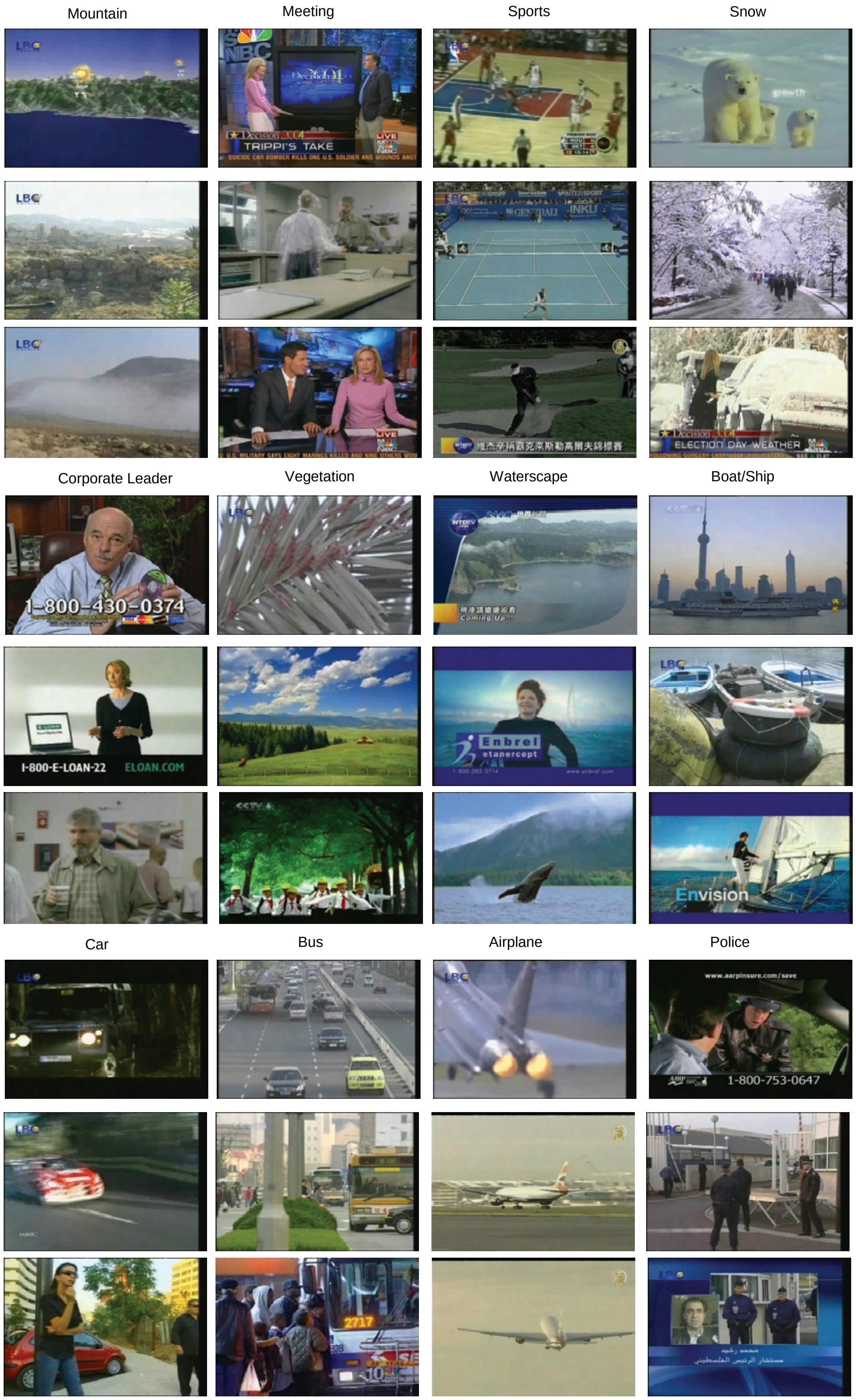


Figure 3.6: Some images corresponding to key frames from training shots provided by TRECVid.

Chapter 4

Scene classification using a hybrid generative/discriminative approach

In this chapter we investigate whether dimensionality reduction using a latent generative model is beneficial for the task of weakly supervised scene classification. In detail we are given a set of labelled images of scenes (e.g. coast, forest, city, river, etc) and our objective is to classify a new image into one of these categories. One approach is to represent each image by a vector, and train a multi-way classifier on these vectors. We compare this approach to that of first discovering latent "topics" using probabilistic Latent Semantic Analysis (pLSA), a generative model from the statistical text literature here applied to a bag of visual words representation for each image, and subsequently training a multi-way classifier on the topic distributions vector for each image.

To this end we introduce a novel vocabulary using dense colour SIFT descriptors, and then investigate the classification performance under changes in the size of the visual vocabulary, the number of latent topics learnt, and the type of discriminative classifier used (k-nearest neighbor or SVM). We achieve superior classification performance to recent publications that have used a bag of visual word representation, in all cases using the authors' own datasets and testing protocols.

4.1 Introduction

Classifying scenes, such as mountains, forests, offices, is not an easy task owing to their variability, ambiguity, and the wide range of illumination and scale conditions that may apply. As was noted in chapter 2, two basic strategies can be found in the literature. The first uses low-level features such as colour, texture, power spectrum, etc. This approach considers the scene as an individual object [134, 143] and is normally used to classify only a small number of scene categories (indoor versus outdoor, city versus landscape etc.). The second strategy uses an intermediate representation before classifying scenes [41, 102, 150], and has been applied to cases where there are a larger number of scene categories (up to 15).

In this chapter we introduce a classification algorithm based on a combination of unsupervised probabilistic Latent Semantic Analysis (pLSA) [65] followed by a discriminative classifier. The pLSA model was originally developed for topic discovery in a text corpus, where each document is represented by its word frequency. Here it is applied to images represented by the frequency of "visual words". The formation and performance of this "visual vocabulary" is investigated in depth. In particular we compare sparse and dense feature descriptors over a number of modalities (colour, texture, orientation). The approach is inspired in particular by three previous papers: (i) the use of pLSA on sparse features for recognizing compact object categories (such as Caltech cars and faces) in Sivic *et al.* [128]; (ii) the dense SIFT [84] features developed in Dalal and Triggs [36] for pedestrian detection; and (iii) the semi-supervised application of Latent Dirichlet Analysis (LDA) for scene classification in Fei-Fei and Perona [41]. We have made extensions over all three of these papers both in developing new features and in the classification algorithm. Our work is most closely related to that of Quelhas *et al.* [116] who also use a combination of pLSA and supervised classification. However, their approach differs in using sparse features and is applied to classify images into only three scene types.

We compare our classification performance to that of four previous methods [41, 76, 102, 150] using the authors' own databases. The previous works used varying levels of supervision in training (compared to the unsupervised topic discovery developed in this work): Fei-Fei and Perona [41] requires the category of each scene to be specified during learning (in order to discover the *themes* (topics) of each category) – we do not specify the category when discovering topics; Oliva and Torralba [102] requires a manual ranking of the training images into 6 different properties; and Vogel and Schiele [150] requires manual classification of 59582 local patches from the training images into one of 9 *semantic concepts*. As will be seen, we achieve superior performance compared to [41, 102, 150]. Lazebnik *et al.* [76] does not use an intermediate topic representation, but improves performance (compared to our approach) by adding spatial information over the bag of words model.

We briefly give an overview of the pLSA model in section 4.2. Then in section 4.3 we describe the hybrid classification algorithm based on applying pLSA to images followed by discriminative classification. Section 4.4 describes the features used to form the visual vocabulary and the principal parameters that are investigated. The used datasets and a detailed description of the experimental procedure evaluation is given in section 4.5. Section 4.6 reports the experimental work performed and main task of investigation. First we optimize the performance over changes in the vocabulary and number of latent topics, then we compare the hybrid classifier to a more standard approach of classifying on the bag of words histograms directly. A comparison with other scene classification methods is given in 4.7. In section 4.8 we demonstrate applications of the hybrid algorithm to relevance feedback, scene classification in videos, and segmentation. In section 4.9 we summarize the properties of the method and discuss its weakness and how they could be solved.

4.2 pLSA model

Probabilistic Latent Semantic Analysis (pLSA) is a generative model from the statistical text literature [65]. In text analysis this is used to discover topics in a document using the bag of words document representation. Here we have *images* as *documents* and we discover *topics as object categories* (e.g. grass, houses), so that an image containing instances of several objects is modelled as a mixture of topics. The models are applied to images by using a *visual* analogue of a *word*, formed by vector quantizing colour, texture and SIFT feature like region descriptors (as described in section 4.4). pLSA is appropriate here because it provides a correct statistical model for clustering in the case of multiple object categories per image. We will explain the model in terms of images, visual words and topics.

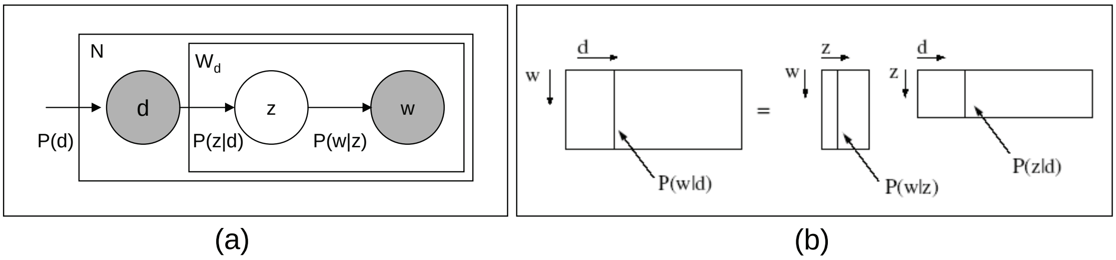


Figure 4.1: (a) pLSA graphical model. Nodes inside a given box (plate notation) indicate that they are replicated the number of times indicated in the top left corner (N=number of images;W_d=number of (visual) words per image). Filled circles indicate observed random variables; unfilled are unobserved. (b) The goal is to find the topic specific word distributions $P(w|z)$ and corresponding document specific mixing proportions $P(z|d)$ which make up the observed document specific word distribution $P(w|d)$.

Suppose we have a collection of images D = d_1,...,d_N with words from a visual vocabulary W = w_1,...,w_V. The data is a $V \times N$ co-occurrence table of counts N_{ij} = $n(w_i, d_j)$, where $n(w_i, d_j,)$ denotes how often the term w_i occurred in an image d_j. A latent variable model associates an unobserved topic variable $z \epsilon Z$ = z_1,...,z_Z with each observation, an observation being the occurrence of a word in a particular image (w_i, d_j). We introduce the following probabilities: $P(d_j)$ denotes the probability of observing a particular image d_j, $P(w_i|z_k)$ denotes the conditional probability of a specific word conditioned on the unobserved topic variable z_k, and finally $P(z_k|d_j)$ denotes an image specific probability distribution over the latent variable space. Using these definitions, the generative model is the following:

- Select an image d_j with probability $P(d_j)$
- Pick a latent topic z_k with probability $P(z_k|d_j)$
- Generate a word w_i with probability $P(w_i|z_k)$.

As a result one obtains an observation pair (w_i, d_j), while the latent topic variable z_k is discarded.

The graphical model representation is shown in figure 4.1a corresponding to a joint probability $P(w, d, z) = P(w|z)P(z|d)P(d)$. Marginalizing out the latent variable z gives:

$$P(w,d) = \sum_{z \epsilon Z} P(w,d,z) = P(d) \sum_{z \epsilon Z} P(w|z)P(z|d) \tag{4.1}$$

and thence from $P(w,d) = P(d)P(w|d)$, we obtain $P(w|d)$ as:

$$P(w|d) = \sum_{z \epsilon Z} P(w|z)P(z|d) \tag{4.2}$$

This amounts to a matrix decomposition as shown in figure 4.1b with the constraint that both the topic vectors $P(w|z)$ and mixture coefficients $P(z|d)$ are normalized to make them probability distributions. Essentially, each image is modelled as a mixture of topics, the histogram for a particular document being composed from a mixture of the histograms corresponding to each topic. In particular each image is a convex combination of the Z topic vectors.

Following the likelihood principle, one determines $P(w|z)$, and $P(z|d)$ by maximization of the loglikelihood function:

$$L = \log P(D,W) = \sum_{d \epsilon D} \sum_{w \epsilon W} n(w,d) \log P(w,d) \tag{4.3}$$

This is equivalent to minimizing the Kullback-Leibler divergence between the measured empirical distribution and the fitted model. The model is fitted using the Expectation Maximization (EM) algorithm as described in [65]. Fitting the model involves determining the topic vectors which are common to all documents and the mixture coefficients which are specific for each document. The goal is to determine the model that gives high probability to the visual words that appear in the corpus.

4.3 Hybrid classification

Training proceeds in two stages. First, the topic specific distributions $P(w|z)$ are learnt from the set of training images. Determining both $P(w|z)$ and $P(z|d_{train})$ simply involves fitting the pLSA model to the entire set of training images. In particular it is not necessary to supply the identity of the images (i.e. which category they are in) or any region segmentation. Each training image is then represented by a Z-vector $P(z|d_{train})$, where Z is the number of topics learnt. In the second stage a multi-class discriminative classifier is trained given the vector $P(z|d_{train})$ of each training image and its class label. For the discriminative stage we compare K Nearest Neighbors classifier (KNN) to a Support Vector Machine classifier (SVM). In more detail, the KNN selects the K nearest neighbors of the new image within the training database (using Euclidean distance). Then it classifies the test image according to the category label which is most represented within the K nearest neighbors. For the SVM classifier an exponential kernel of the form $\exp -\alpha d$ is used, where d is the Euclidean distance between the vectors, and the scalar α is determined as described in [158] (we use the LIBSVM package [31] with the trade-off between training error and margin at $C = 1$). The multi-way classification is done using the one-versus-all rule: a classifier is learned to separate each class from the rest, and a test image is assigned the label of the classifier with the highest response.

Classification of an unseen test image similarly proceeds in two stages. First the document specific mixing coefficients $P(z|d_{test})$ are computed, and these are then used to classify the test images using a discriminative classifier. In more detail document specific mixing coefficients $P(z|d_{test})$ are computed using the fold-in heuristic described in [64]. The unseen image is projected onto the simplex spanned by the $P(w|z)$ learnt during training, i.e. the mixing coefficients $P(z_k|d_{test})$ are sought such that the Kullback-Leibler divergence between the measured distribution and $P(w|d_{test}) = \sum_{z \epsilon Z} P(w|z)P(z|d_{test})$ is minimized. This is achieved by running EM in a similar manner to that used in learning, but now only the coefficients $P(z_k|d_{test})$ are updated in each M-step with the learnt $P(w|z)$ kept fixed. The result is that the test image is represented by a Z-vector. The test image is then classified by the multi-class discriminative classifier (KNN or SVM) as described above. Figure 4.2 shows graphically the hybrid generative/discriminative process for both training and testing.

4.4 Visual words and visual vocabulary

In the formulation of pLSA, we compute a co-occurrence table, where each image is represented as a collection of visual words, provided from a visual vocabulary. This visual vocabulary is obtained by vector quantizing descriptors computed from the training images using k-means, see the illustration in the first part of figure 4.2. Previously both

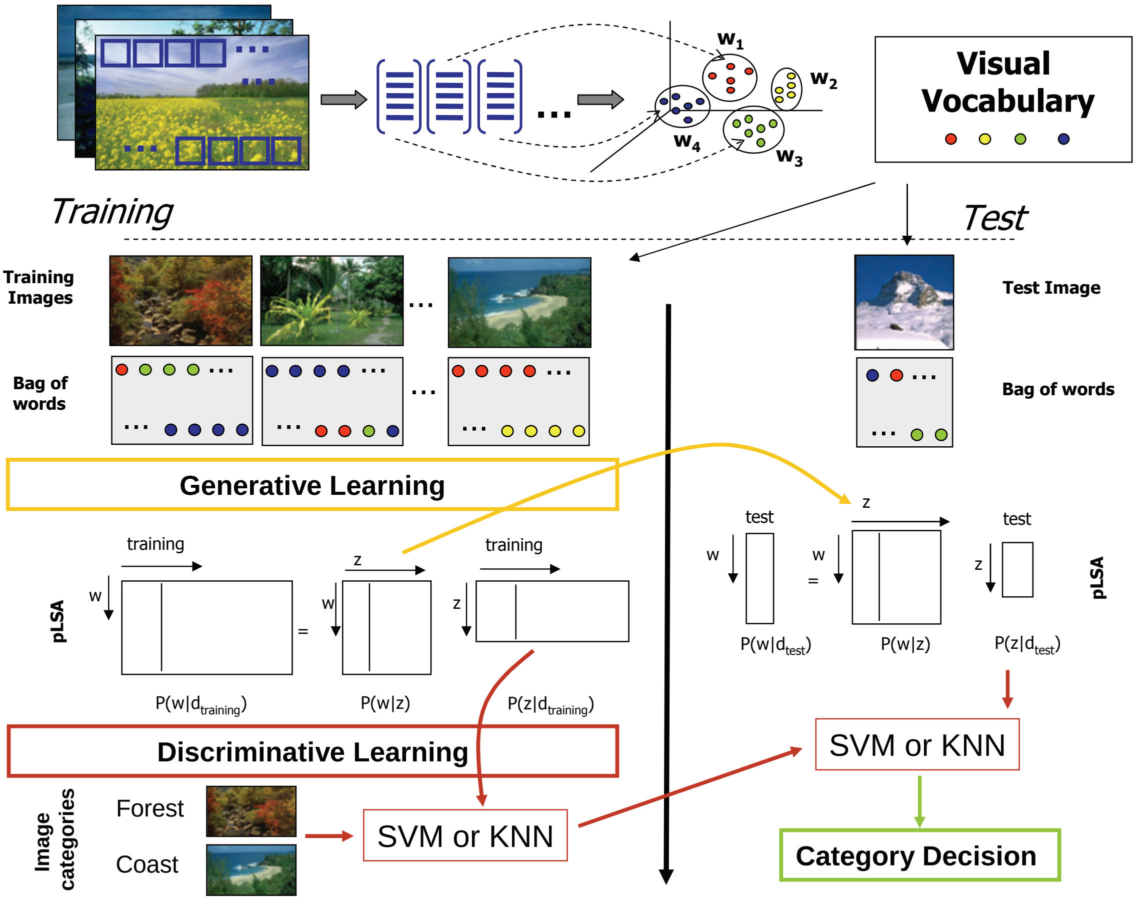


Figure 4.2: Overview of visual vocabulary formation, learning and classification stages.

sparse [35, 75, 129] and dense descriptors, e.g. [36, 79, 147], have been used. Here we carry out a thorough comparison over dense descriptors for a number of visual measures (see below) and compare to a sparse descriptor. We vary the size of the patches and degree of overlap, and compare normalized to unnormalized images. We then assess classification performance over four different image datasets described in section 4.5.

We investigate four dense descriptors, and compare their performance to a previously used sparse descriptor. In the dense case the important parameters are the size of the patches (N) and their spacing (M) which controls the degree of overlap:

- **Grey patches** (dense). As in [147], and using only the grey level information, the descriptor is a N × N square neighborhood around a pixel. The pixels are row reordered to form a vector in an N^2 dimensional feature space. The patch size tested are $N = 5$, 7 and 11. The patches are spaced by M pixels on a regular grid.

The patches do not overlap when $M = N$, and do overlap when $M = 3$ (for N = $5, 7$) and M = 7 (for N = 11).

- **Colour patches** (dense). As above, but the colour information is used for each pixel. We consider the three colour components HSV and obtain a $N^2 \times 3$ dimensional vector. As in the grey level, we used N = 5, 7, and 11. We use HSV because of its similarities to the way humans tend to perceive colour and because it is less sensitive to shadow and shading.

- **Grey SIFT** (dense). SIFT descriptors [84] are computed at points on a regular grid with spacing M pixels, here M = 5, 10 and 15. At each grid point SIFT descriptors are computed over circular support patches with radii r = 4, 8, 12 and 16 pixels. Consequently each point is represented by n SIFT descriptors (where n is the number of circular supports), each is 128-dim. Multiple descriptors are computed to allow for scale variation between images. The patches with radii 8, 12 and 16 overlap. Note, the descriptors are rotation invariant.

- **Colour SIFT** (dense). As above, but now SIFT descriptors are computed for each HSV component. This gives a 128×3 dim-SIFT descriptor for each point. Note, this is a novel feature descriptor. It captures the colour gradients (or edges) of the image. Other ways of using colour with SIFT features have been proposed by [53, 152].

- **Grey SIFT** (sparse). Affine co-variant regions are computed for each grey scale image, constructed by elliptical shape adaptation about an interest point [93]. These regions are represented by ellipses. Each ellipse is mapped to a circle by appropriate scaling along its principal axis and a 128-dim SIFT descriptor computed. This is the method used by [35, 75, 128, 129].

4.4.1 Implementation details

Dense SIFT descriptors

In most previous applications SIFT like descriptors are used following a sparse feature detection, and so have only been applied at image points where there is sufficient structure (e.g. a strong response from a Harris or Hessian operator). In our case the SIFT descriptors are applied densely, perhaps at every pixel, and this raises two areas of concern.

First, in regions with near constant colour/brightness (like sky, road) that consequently have small image gradients, is the resulting description (the visual words) very sensitive to noise? In practice we find that the assigned word for such patches is often the same and relatively insensitive to patch size. For example if sky patches with $r = 4$ are assigned the word w_1, then sky patches with $r = 8$ are also assigned the word w_1 and so on. Where the small gradients (noise) do result in different random visual word assignments, then the pLSA topic learns this distribution.

Second, is there a problem with noise causing wrap-around in the H colour channel? This could occur with a region consisting of small fluctuations around saturated red, and would result in an alternation of visual word assignment over that region. However, in practice we do not observe this problem in the current databases.

Normalization

Grey level images are normalized to have intensities with mean zero and unit standard deviation. Colour images are first normalized as in “Gray World” [34, 47] to have R,G and B components $R * (\mu/\mu_r), G * (\mu/\mu_g), B * (\mu/\mu_b)$ where $\mu = (\mu_r + \mu_g + \mu_b)/3$ and μ_r, μ_g, μ_b are the mean of each component. The HSV is then computed from these normalized values.

4.5 Datasets and methodology

Datasets

We evaluated our classification algorithm on four different datasets: (i) Oliva and Torralba [102], (ii) Vogel and Schiele [150], (iii) Fei-Fei and Perona [41] and Lazebnik et *al* [76]. As was noted in chapter 3, we will refer to these datasets as OT, VS, FP and LSP respectively. All these four datasets consist of images from natural and man-made scenes and are described in detail in chapter 3.

Methodology

The classification task is to assign each test image to one of a number of categories. The performance is measured using a confusion table, and overall performance rates are measured by the average value of the diagonal entries of the confusion table.

Datasets are split randomly into two separate sets of images, half for training and half for testing. From the training set we randomly select 100 images to form a validation set. This validation set is used to find the optimal parameters, and the rest of the training images are used to compute the vocabulary and pLSA topics. A vocabulary of visual words is learnt from about 30 random training images of each category.

Excluding the preprocessing time of feature detection and visual vocabulary formation, it takes about 20 mins to fit the pLSA model to 1600 images (Matlab implementation on a 1.7GHz computer).

The new classification scheme is compared to two baseline methods. These are included in order to gauge the difficulty of the various classification tasks. The baseline algorithms are:

Global colour model. The algorithm computes global HSV histograms for each training image. The colour values are represented by a histogram with 36 bins for *H*, 32 bins for *S*, and 16 bins for *V*, giving a 84-dimensional vector for each image. A test image is classified using KNN (with $K = 10$).

Global texture model. The algorithm computes the orientation of the gradient at each pixel for each HSV channel at each training image. These orientations are collected into a 72 bin histogram for each colour channel and concatenated to form a histogram of 72×3 bins for each image. The classification of a test image is again carried out using KNN.

4.6 Classification results

In this section we carry out a set of experiments to investigate the various choices of vocabularies, parameters and classifiers, and also to assess the benefits or otherwise of using pLSA as an intermediate representation.

The experiments in this section are all on the OT dataset. The results for the other datasets (FP, VS and LSP) are given in section 4.7. For the OT dataset three classification situations are considered: classification into 8 categories, and also classification within the two subsets of natural (4 categories), and man-made (4 categories) images. The latter two are the situations considered in [102].

We start by finding the optimal parameters (V, Z and K) over the validation set for each of the different vocabularies described in section 4.6.1. The optimal parameters are then fixed, and subsequent results reported on the test set.

4.6.1 Optimizing the parameters V, Z and K (on the validation set)

We first investigate how classification performance (on the validation set) is affected by the various parameters: the number of visual words (V in the k-means vector quantization), the number of topics (Z in pLSA), and the number of neighbors (K in kNN). Figure 4.3 shows this performance variation for two types of descriptor – dense colour SIFT with $M = 10$ and four circular supports, and grey patches with $N = 5$ and $M = 3$. Note the mode in the graphs of V, Z and K in both cases. This is quite typical across all types of visual words, though the position of the modes vary slightly. For example, using colour SIFT the mode is at $V = 1500$ and $Z = 25$, while for grey patches the mode is at $V = 700$ and $Z = 23$. For K the performance increases progressively until K is between 7 and 12, and then drops off slightly.

For colour patches the best performance is obtained when using the 5×5 patch over normalized images, with M = 3, V = 900, Z = 23 and K = 10. The best results overall are obtained with dense colour sift with 4 circular supports, $M = 10$, normalized images, V = 1500, Z = 25 and K = 10. We will see in next section that this vocabulary is also the one which gives the best results on the test set.

4.6.2 Comparison of features and support regions (on the test set)

We next investigate the patch descriptors in more detail. Again, we use the OT dataset with 8 categories and the KNN classifier for this task (the SVM classifier is investigated in section 4.6.3). In the following results the optimum choice of parameters determined on the validation set is used for each descriptor type, but here applied to the test set.

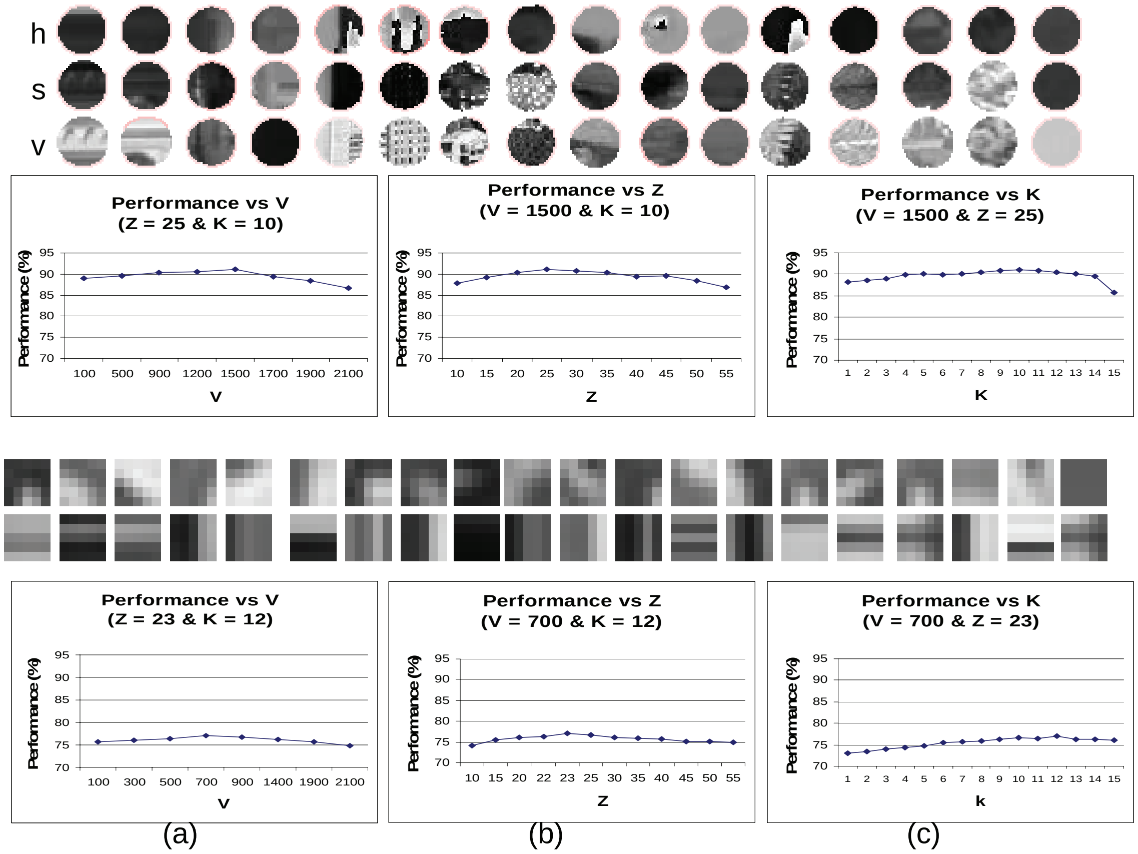


Figure 4.3: Validation set performance under variation in various parameters for the 8 category OT classification. Top: example visual words and performance for dense colour SIFT $M = 10$, $r = 4$, 8, 12 and 16 (each column shows the HSV components of the same word). Lower example visual words and performance for grey patches with $N = 5$ and $M = 3$. (a) varying number of visual words, V, (b) varying number of topics, Z, (c) varying k (KNN).

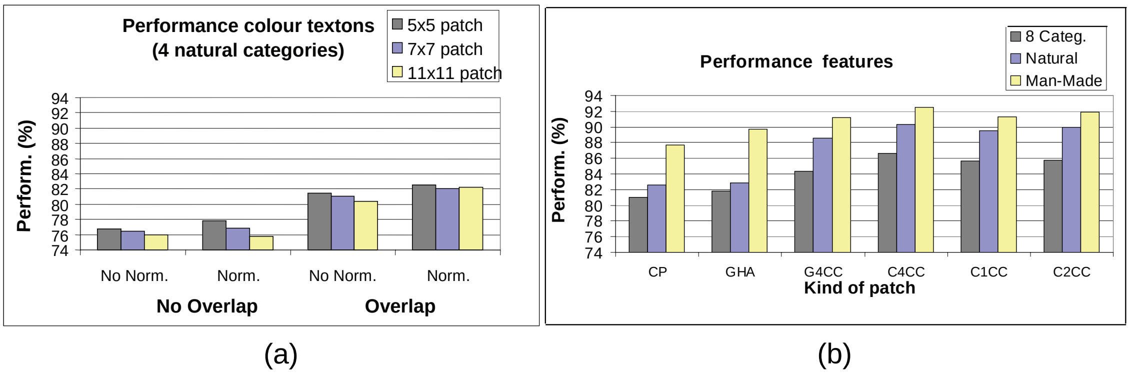


Figure 4.4: (a) The performance when classifying the four natural categories using normalized and unnormalized images and with overlapping and non-overlapping patches. Colour patches are used. (b) Performance when classifying all categories, man-made and natural using different patches and features. Abbreviations for this and subsequent figures: CP (Colour Patches), GHA (Grey Harris Affine – sparse, all the other descriptors are dense), G4CC (Grey SIFT four Concentric Circles), CxCC (Colour SIFT with x Concentric Circles).

Figure 4.4a shows the results when classifying the images of natural scenes with colour-patches. The performance when using normalized images is nearly 1% better than when using unnormalized. When using overlapping patches, the performance increases by almost 6% compared to no overlap. Similar results occur for the man-made and all scene category sets. Comparing results when classifying the images using only grey level information or using colour, it can be seen in figure 4.4b and table 4.2, that colour brings an increment of around 2%. This is probably because colour is such an important factor in outdoor images, and helps to disambiguate and classify the different objects in the scene.

The performance of SIFT features is shown in figure 4.4b. The best results are obtained with dense and not sparse descriptors. This is almost certainly because we have more information on the images: in the sparse case the only information is where a Harris detector fires and, especially for natural images, this is a very impoverished representation (see figure 2.8b). Again colour is a benefit with better results obtained using colour than grey SIFT. The performance using grey SIFT when classifying natural images is 88.5% and increase 2% when using colour SIFT, both with four concentric support regions. The difference when using these vocabularies with man-made images is not as significant. This reiterates that colour in natural images is very important for classification.

Training Regions	Testing Regions		
	4CC	2CC	1CC
4CC	86.9	86.7	86.3
Same as Testing	86.9	85.8	85.7

Table 4.1: Changing the number of training and test support regions (for OT 8 categories). First row: each pixel in the training images are represented by four circles (4CC) and the testing images are represented by four (4CC), two (2CC) and one (1CC) circle from left to right. Second row: pixels in the training and testing images are represented by the same number of circles.

Number of support regions

Turning to the performance variation with the number of support regions for dense SIFT. It can be seen from figure 4.4b that best results are obtained using four concentric circles. With only one support region to represent each patch, results are around 1% worse. This is probably because of lack of invariance to scale changes: using four support regions to represent each pixel effectively represents the texture at four different scales.

We now investigate how important it is to use four concentric circles to represent each pixel in both training *and* testing. The first row of table 4.1 shows the performance when using four concentric circles with colour to represent each pixel at the training stage, and four, two and one circles also with colour information for the testing data. The second row shows the performances when using the same number of circles to represent the pixels at the training and testing stage. It can be seen that performances in the first row are very similar, so that four concentric circles is enough to represent the training data and fewer patches can be used to represent the pixels in the testing images, i.e. sampling only the training images at multiple scales is sufficient.

Table 4.2 summarizes the results for the three OT image sets (all 8 categories, 4 natural and 4 man-made) covering the different dense vocabularies: grey and colour patches, grey and colour SIFT and the two baseline algorithms when using KNN classifier. From these results it can be seen that: (i) The baseline texture algorithm works better than the baseline colour in all three cases. Despite its simplicity the performance of the baseline texture algorithm on man-made images (73.8%) is very high, showing that these images may be easily classified from their edge directions. (ii) For the various descriptors there

Visual Vocabulary	GP	CP	G4CC	C4CC	PS	GlC	GlT
All categ.	71.5	77.0	84.3	**86.6**	82.6	55.1	64.6
Natural categ.	75.4	82.4	88.5	**90.2**	84.0	59.5	70.1
Man-made categ.	77.4	83.5	91.1	**92.5**	89.3	66.1	73.8

Table 4.2: Rates obtained different features when using database OT. GP (Gray Patches), CP (Colour Patches), G4CC (Grey SIFT four Concentric Circles), C4CC (Colour SIFT with four Concentric Circles). PS (Colour Patches and Colour SIFT), GlC (Global Colour), GlT (Global Texture).

	pLSA		BoW	
	KNN	SVM	KNN	SVM
C4CC	86.6	**87.1**	82.5	83.8
G4CC	84.3	84.7	79.7	80.8

Table 4.3: Rates obtained for KNN and SVM when data provided by pLSA and BoW are used as input vectors for the classifiers. OT database (8 categories) is used. G4CC (Grey SIFT four Concentric Circles), C4CC (Colour SIFT with four Concentric Circles)

are clear performance conclusions: man-made is always better classified than natural (as expected from the baseline results); SIFT type descriptors are always superior to patches; colour is always superior to grey level. The best performance (86.6% for all 8 categories) is obtained using colour SIFT and four concentric circles. (iii) Somewhat surprisingly, better results are obtained using the SIFT vocabulary alone, rather than when merging both vocabularies (patches and SIFT). This may be because the parameters (V, Z and K) have been optimized for a single vocabulary, not under the conditions of using multiple vocabularies.

4.6.3 KNN vs SVM

All the results above are for $P(z|d)$ with the KNN classifier. Now we investigate classification performance when using a SVM. Table 4.3 shows the results for the SIFT support regions for both classifiers KNN and SVM. Optimized parameters for each vocabulary are used. It can be seen that SVM performs around 1% better than KNN.

4.6.4 pLSA vs Bag-of-Words (BoW)

The results to this point use pLSA to obtain an intermediate representation, with $P(z|d)$ as the inputs for the classifiers. We now compare to the performance obtained by classifying the BoW representation directly. Again the performance is for the OT dataset with 8 categories, and in all the experiments: $V = 1500$ (unless stated otherwise), $Z = 25$, $K = 10$, and four support regions are used for each point spaced at $M = 10$. For the SVM classifier a χ^2 exponential kernel [158] is used for the BoW, and an Euclidean exponential kernel for pLSA. These kernels were found to give the best performance in each case.

Table 4.3 shows pLSA and BoW rates for different support regions and using a SVM and KNN. It can be seen that in all cases the performance using pLSA is around 4% better than that obtained using a BoW.

Number of training Images

We now evaluate the classification performance when less training data is available. The OT dataset is split into 2000 training images and 688 test images. A varying number, n_{train}, of images from the training set are used for both learning the pLSA topics (generative part) and learning the topic distribution of each scene (discriminative part). The classification performance using $P(z|d)$ is compared to that of using BoW vectors. As can be seen in figure 4.5, the gap between pLSA and BoW increases as the number of labelled training images decreases, as was demonstrated in [116].

In the previous experiment, we varied the amount of training data for both: the generative and discriminative learning. However, a key advantage of the hybrid approach is that the generative part of the model can be trained on large amounts of unlabelled data (hence discovering the structure of the data), so that relatively few labelled examples are needed for high accuracy. To show this advantage, we repeat the previous experiment training the generative classifier using the 2000 training images and decreasing the number of labelled training images (n_{train}) only for the discriminative classifier. Figure 4.6 shows the comparison of the previous experiment and the current experiment when using SVM as a discriminative classifier. It can be seen that much better results are obtained when decreasing only the number of labelled training data than when reducing the training data

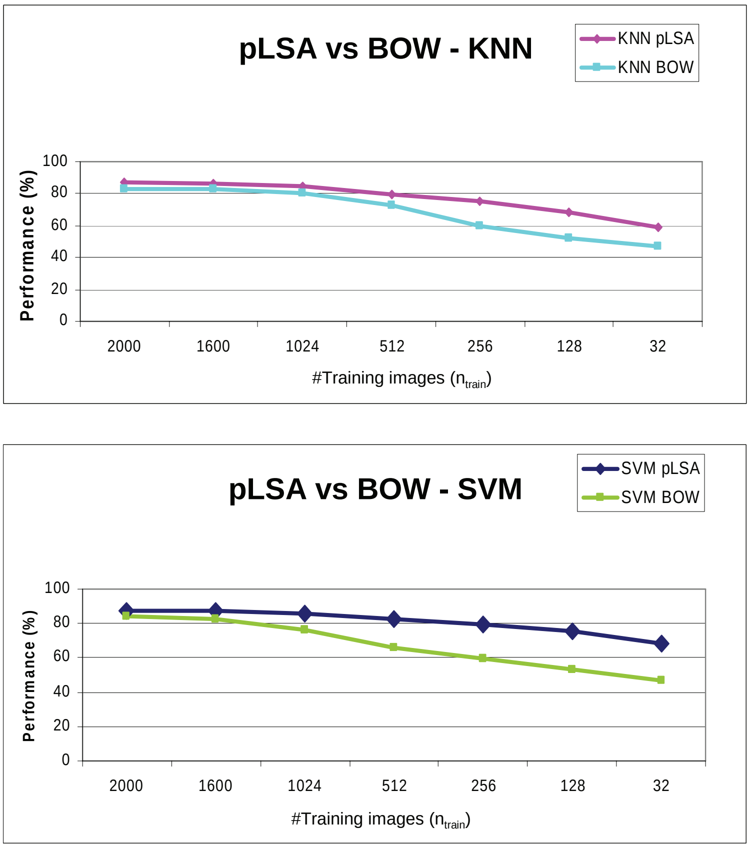


Figure 4.5: pLSA and BoW performances when decreasing the number of training images. 8 categories from the OT dataset with four concentric circles and $V = 1500$ words, $Z = 25$ and $K = 10$.

in both learning parts. So there is a clear advantage of using a hybrid approach: the system has acceptable performances with less labelled training data.

Vocabulary size

Figure 4.7 shows the performance when changing the vocabulary size V (from 200 to 5000 words) for both the discriminative classifiers (KNN and SVM). It can be seen that for both classifiers, pLSA is less affected by the vocabulary size than the BoW.

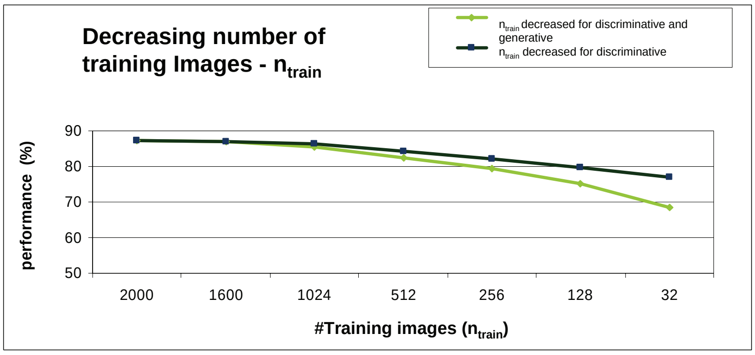


Figure 4.6: System performances when decreasing the number of training images in the generative and discriminative approach (blue line) and when deceasing the number of labelled training images only for the discriminative approach (yellow line). 8 categories from the OT dataset with four concentric circles and $V = 1500$ words, $Z = 25$ and $K = 10$. SVM is used as a discriminative classifier.

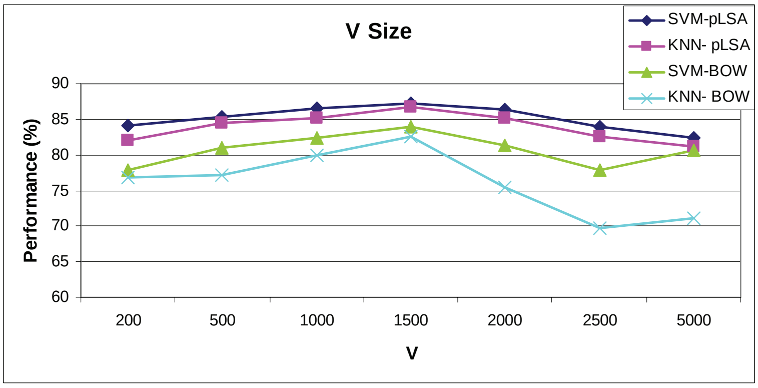


Figure 4.7: Changing the vocabulary size for the OT dataset. Parameters are $Z = 25$, $K = 10$, $M = 10$ and four concentric circles.

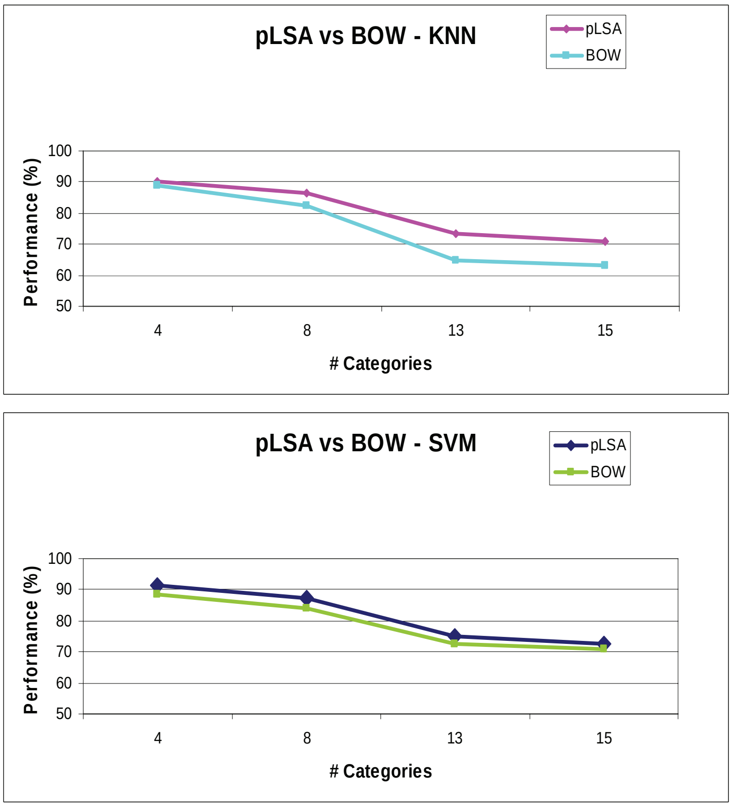


Figure 4.8: pLSA and BoW performances when classifying different number of categories (from 4 to 15). Parameters used are V = 1500, Z = 25, $M = 10$ and 4 concentric circles as support regions. First row: pLSA vs BoW when using KNN ($K = 10$); second row: pLSA vs BoW when using SVM.

Number of scene categories

Figure 4.8 shows the performances when increasing the number of categories to be classified for both KNN (first row) and SVM (second row). For the KNN, when classifying the 4 natural images in the OT dataset, the results using the topic distribution is 90.2% and with the BoW directly the classification performance decreases by only around 1.5%, to 88.7%. However for 8 categories, the performance decreases by nearly 4%, from 86.6% to 82.5%. Using the 13 categories from the FP dataset and the 15 LSP dataset, the performance falls around 8%, from 73.4% to 64.8% and from 71.0% to 63.1% respectively. Thus there is a clear gain in using pLSA (over the BoW) with KNN when classifying a large number of categories.

	KNN		SVM	
# Categ.	pLSA	BoW	pLSA	BoW
4 OT dataset	90.2	88.7	**91.5**	88.4
8 OT dataset	86.6	82.5	**87.1**	83.8
13 FP dataset	73.4	64.8	**74.9**	73.6
15 LSP dataset	71.0	63.1	**72.6**	72.5

Table 4.4: Classification rates for pLSA and BoW when classifying categories from different datasets. Parameters used are V = 1500, Z = 25, $M = 10$ and 4 concentric circles as support regions.

If we focus on the SVM, performances with pLSA are better as well. However when classifying a large number of categories (13 or 15) pLSA is 1% better than BoW, thus the gap is not as large as when using the KNN classifier. Table 4.4 summarizes the performances for KNN and SVM over pLSA and BoW.

4.6.5 Summary

The best results are obtained using dense descriptors – colour SIFT with four circular support regions. Overlap increases the performance. When using the SIFT vocabulary the values for the parameters giving the best results are $M = 10$ pixels with concentric circles support regions of $r = 4$, 8, 12 and 16 pixels. For patches the best results are for $N = 5$, $M = 3$. Table 4.5 shows the optimized values V, Z and K learnt from a validation set for each dataset. Note that V strongly depends on the size of the feature vector (128×3 dimensionality vector for SIFT and 25×3 dimensionality vector for patches), while Z depends on the number of categories in each dataset. In both (SIFT and patches), colour information increases performance. The result that dense SIFT gives the best performance was also found by [36] in the case of pedestrian detection. It it interesting that the same feature applies both to more distributed categories (like grass, mountains) as well as the compact objects (pedestrians) of their work where essentially only the boundaries are salient.

When comparing the discriminative classifiers KNN and SVM, better performances are obtained with SVM. We also demonstrated that pLSA works better than the BoW

	SIFT			Patch		
Dataset	V	Z	K	V	Z	K
VS	1500	25	7	900	25	9
OT	1500	25	10	900	23	10
FP	1200	35	9	600	33	10
LSP	1200	40	11	700	42	12

Table 4.5: Optimized parameters when using the SIFT vocabulary for the four datasets: $M = 10$ and $r = 4$, 8, 12 and 16 pixels, and when using the patch vocabulary: $N = 5$, $M = 3$ pixels. A validation set is used for each dataset.

representation (pLSA provides a better intermediate representation of the images), and that pLSA is less affected by the vocabulary size and the number of training images. More concretely for the KNN discriminative classifier, when working with a small number of categories the difference between pLSA and BoW is 1.5%. However when the number of categories increases this difference is around 8% showing that pLSA provides a more robust intermediate representation than BoW. Thus there is a clear gain in using pLSA (over the BoW) with KNN when classifying a large number of categories. Moreover a clear advantage of using a generative model (pLSA), over BoW directly, is that the number of *labelled* training images can be reduced considerably without much loss of performance.

Moreover a clear advantage of using a generative classifier (pLSA) over BoW to discover the topic representation of the images is that the number of labelled training images can be considerably reduced without much loss of performance. We can train the system with a large amount of unlabelled images and use few labelled images reducing the human annotation effort.

4.7 Comparison to previous results

We compare the performance of our scene classification algorithm to the supervised approaches of Vogel and Schiele [150] and Oliva and Torralba [102], and the semi-supervised approach of Fei-Fei and Perona [41] and Lazebnik et *al.* [76], using the same datasets (VS, OT, FP and LSP respectively) that they tested their approaches on and the

Dataset	# of categ.	# train	# test	hybrid approach	Authors
OT	8	800	1888	83.5 ± 1.2	**83.7** [1]
OT	4 Natural	1000	472	**90.7** ± 0.9	89.0 [102]
OT	4 Man-Made	1000	216	**91.7** ± 1.2	89.0 [102]
VS	6	600	100	**87.8** ± 1.0	74.1 [150]
FP	13	1300	2459	**74.3** ± 1.3	65.2 [41]
LSP	15	1500	2986	72.7 ± 1.2	**81.4** [76]

Table 4.6: Comparison of our algorithm with other methods using their own databases. Validation set optimized values are used for each dataset.

same number of training and testing images. For each dataset we use the SVM classifier, SIFT and four circular supports spaced at $M = 10$; the parameters V and Z have the optimized values for each dataset (see table 4.5). We used colour for OT and VS, and grey for FP and LSP. The visual vocabulary is computed independently for each dataset, as described in section 4.5. We return to the issue of sharing vocabularies across datasets in section 4.8. The results are given in table 4.6.

Note that much better results are obtained with the four natural scenes of OT, than with the six of VS. This is because the images in VS are much more ambiguous than those of OT and consequently more difficult to classify. In table 4.6 we can see that our method outperforms the previous methods in [41, 102, 150], despite the fact that our training is unsupervised in the sense that the scene identity of each image is unknown at the pLSA stage and is not required until the SVM training step. This is in contrast to [41, 76], where each image is labelled with the identity of the scene to which it belongs during the training stage. In [150], the training requires manual annotation of 9 semantic concepts for 60000 patches, while in [102] training requires manual annotation of 6 properties for thousands of scenes. It is worth to state that in [102, 150] the intermediate information which represents the images has a semantic meaning while in [41, 76] and our approach the intermediate information has not a semantic meaning from the human point of view. However this is not a problem, because our final goal is to give a label for each scene: we are interested in the semantic meaning of the whole scene and not in the semantic meaning of the intermediate information. In the case of the 8 categories in OT the method

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/

in [102] is only marginally better (0.2%) than ours. The method in [76] is nearly 10% better than ours.

Discussion

The superior performance (compared to [41, 150]) could be due to the use of better features and how they are used. In the case of Vogel and Schiele [150], they learn 9 topics (called *semantic concepts*) that correspond to those that humans can observe in the images: *water*, *trees*, *sky* etc. for 6 categories. Fei-Fei and Perona learn 40 topics (called *themes*) for 13 categories. They do not say if these topics correspond to natural objects. In our case, we discover between 22 and 30 topics for 8 categories. These topics can vary depending if we are working with colour features (where topics can distinguish objects with different colors like *light sky*, *blue sky*, *orange sky*, *orange foliage*, *green foliage* etc...) or only grey SIFT features (objects like *trees* and *foliage*, *sea*, *buildings* etc...). In contrast to [150] we discover objects that sometimes would not be distinguished in a manual annotation, for example *mountains with snow* and *mountains without snow*. Our superior performance compared to [102], when using the 4 natural or man-made categories, could be due to their method of scene interpretation. They use the spatial envelope modelled in a holistic way in order to obtain the structure (shape) of the scene using coarsely localized information. On the other hand, in our approach specific information about objects is used for scene categorization. Performance in [102] is 0.2% better when using the 8 categories in the OT dataset. This score has been obtained with few training images (100 per category in contrast to the 4 natural/man-made where 250 per category have been used). Our method is better if more images are used in the generative learning (as it is demonstrated for the 4 Natural and 4 Man-Made images – see table 4.6), because it is an unsupervised approach and needs more information to discover the class specific distributions. Nevertheless, this performance has been overcome by our approach, with a 87.1%, when more training images are used (see table 4.4).

We do not outperform the spatial pyramid classifier proposed in [76]. This is because their spatial representation is more discriminative for the scene classification task. However when using their method without spatial information their performance is 72.2% and we increase up to 72.45% (here we used $V = 200$ for a better comparison). This means that if we find a way to codify the spatial information into the proposed hybrid approach, we can also improve on performance. This issue will be addressed in chapter 6.

4.8 Applications

We applied the pLSA based classifier in four other situations. The first one is also a classification task, but combining the images of two different datasets, the second is a relevance feedback application, the third is scene retrieval for the film *Pretty Woman* [Marshall, 1990], and in the fourth we apply pLSA for image segmentation. In all the following the descriptor is dense colour SIFT with circular support and $V = 700$, $Z = 22$ and $K = 10$ (these are the optimal parameter values when working with the four natural scenes from the OT dataset).

Vocabulary generalization

In this classification test, we train the system with the four natural scenes of the OT dataset (*coast*, *forest*, *mountains* and *open country*) and test using the same four scene categories from the VS dataset. This tests whether the vocabulary and categories learnt from one dataset generalize to another. We obtain a performance of 88.2% of correctly classified images for KNN and 88.9% for SVM. This performance is only slightly worse than the 89.8% obtained when classifying the same four categories in the VS dataset with no generalization (i.e. using training images only from VS). This slight performance drop is because (i) images within the same database are more similar, and (ii) the images in VS are more ambiguous than OT, so this ambiguity is not represented in training the OT classifier. However, 88.9% compared to 89.8% does demonstrate excellent generalization. To address (i) we investigate using a vocabulary composed from both databases and find this improves the performance to 89.6%.

Relevance Feedback (RF)

[159] proposed a method for improving the retrieval performance, given a probabilistic model. It is based on moving the query point in the visual word space towards good example points (relevant images) and away from bad example points (irrelevant images).

The vector moving strategy uses the Rocchio's formula [117]:

$$q_{pos} = \alpha q + \beta(\frac{1}{a}\sum_{i=1}^{a} rel_i) - \gamma(\frac{1}{b}\sum_{j=1}^{b} irel_j) \tag{4.4}$$

where q is the BoW for the query image, a is the number of relevant images b is the number of irrelevant images, and rel, $irel$ are the BoW representations for the relevant and irrelevant retrieved images. The parameters α, β and γ are set to 1. With the modified query vector q_{pos} and a constructed negative example q_{neg}:

$$q_{neg} = \alpha(\sum_{j=1}^{b} irel_j) + \beta(\frac{1}{b}\sum_{j=1}^{b} irel_j) - \gamma(\frac{1}{a}\sum_{i=1}^{a} rel_i) \tag{4.5}$$

their representations in the discovered concept space are obtained $P(z|q_{pos})$ and $P(z|q_{neg})$ and their similarities sp_i and sn_i to each image $i\epsilon I$ in the database are measured using the cosine metric of the corresponding vectors in the topic space, respectively. Then the images are ranked based on the similarity $s_i = sp_i - sn_i$.

To test RF we simulate the user's feedback using 25 random images of each category. For each query image, we carry out *n* iterations. At each iteration the system examines the top 20, 40 or 60 images that are most similar to the query excluding the positive examples labelled in previous iterations (this is refered as P(20), P(40) and P(60) respectively).Images from the same category as the initial query will be used as positive examples, and other images as negative examples.

Figure 4.9a shows the average precision vs recall graph for all 8 categories in the OT dataset. The cyan line represents the performance in the first iteration (Content Based Image Retrieval CBIR) and the others when using the RF algorithm after 4 iterations. Best results are obtained when considering the top 60 images (P(60)). Figure 4.9b shows the results for 200 query images, 25 of each category, in OT considering P(60) also after 4 iterations. The first 100 images can be retrieved with an average precision of 0.75. We can see that the most difficult category to retrieve is *open country* while the better retrieved are *forest* and *highway* followed by *tall buildings*. This is in accordance with the classification results.

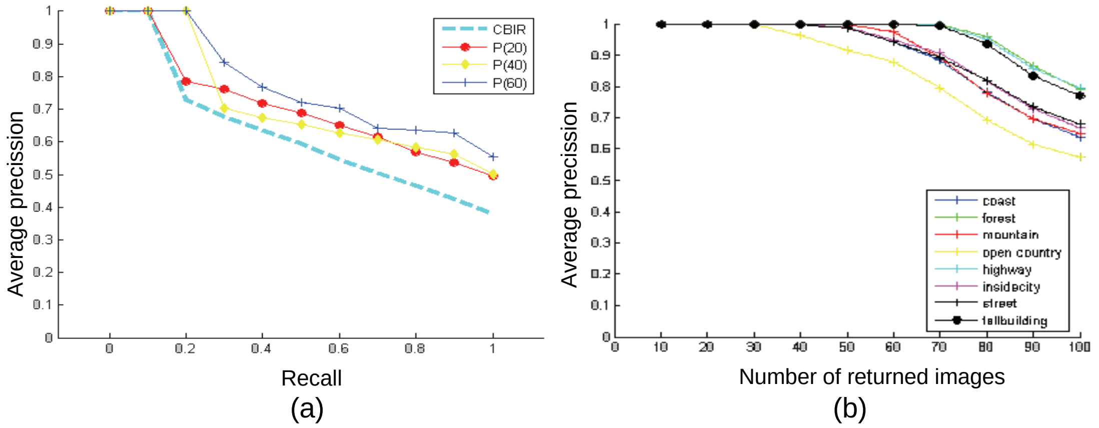


Figure 4.9: (a) Precision vs Recall graph. (b) Performance when retrieving each category separately (25 images query for each).

Classifying film frames into scenes

In this test the images in OT are again used as training images (8 categories), and key frames from the movie *Pretty Woman* are used as test images. We used $V = 1500$ and $Z = 25$ which are the optimized values for the 8 categories in the OT dataset. Note, this is a second example of vocabulary and topic generalization as we are using training images from a different dataset. We used every hundredth frame from the movie to form the test set. In this movie there are only a few images that could be classified as the same categories used in OT, and there are many images containing only people. So it is a difficult task for the system to correctly classify the key frames. Although the results obtained (see figure 4.10) are purely anecdotal, they are very encouraging and show again the success of using pLSA in order to classify scenes according to their topic distribution.

Segmentation

Figure 4.11 shows examples of segmentation of five topics using the colour SIFT vocabulary. Circular patches are painted according to the maximum posterior $P(z|w, d)$:

$$P(z|w,d) = \frac{P(w|z)P(z|d)}{\sum_{z_l \epsilon Z} P(w|z_l)P(z_l|d)} \quad (4.6)$$

For each visual word in the image we choose the topic with maximum posterior $P(z|w, d)$ and paint the patch with its associated colour, so each colour represents a dif-

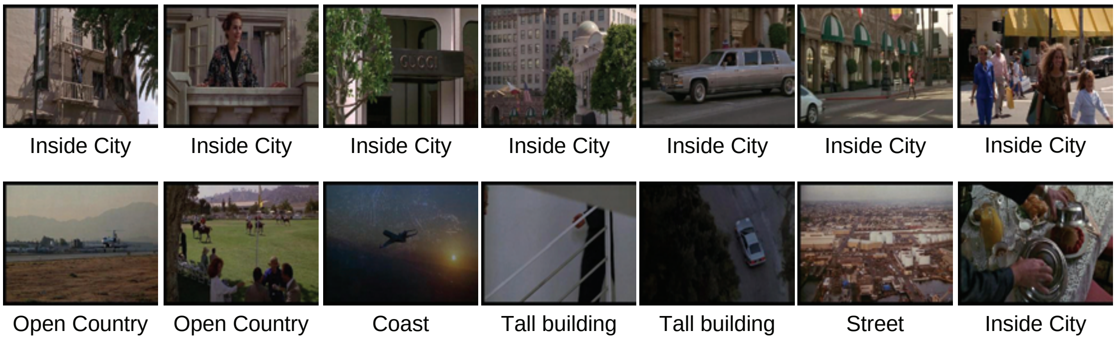


Figure 4.10: Example frames from the film Pretty Woman with their classification. The classifier is trained on the OT dataset.

ferent topic (the topic colour is chosen randomly). To simplify the figures we only paint one topic each time. Note that topics represent consistent regions across images (enabling a coarse segmentation) and there is a straightforward correspondence between topic and object.

4.9 Discussion

We have proposed a scene classifier that learns topics and their distributions in unlabelled training images using pLSA, and then uses their distribution in test images as a feature vector in a supervised discriminative classifier. We have shown the advantage of the hybrid approach when decreasing the number of labelled training images in the training step.

We studied the influence of various descriptor parameters and have shown that using colour (if available) dense SIFT descriptors with overlapping patches gives the best results for man-made as well as for natural scene classification. Furthermore, discovered topics correspond fairly well with different textural objects (*grass*, *mountains*, *sky*) in the images, and topic distributions are consistent between images of the same category. It is probably this freedom in choosing appropriate topics for a dataset, together with the optimized features and vocabularies, that is responsible for the superior performance of the scene classifier over previous work (with manual annotation and without spatial information). Moreover, the use of pLSA is never detrimental to performance, and it gives a significant improvement over the original BoW model when a large number of scene categories are

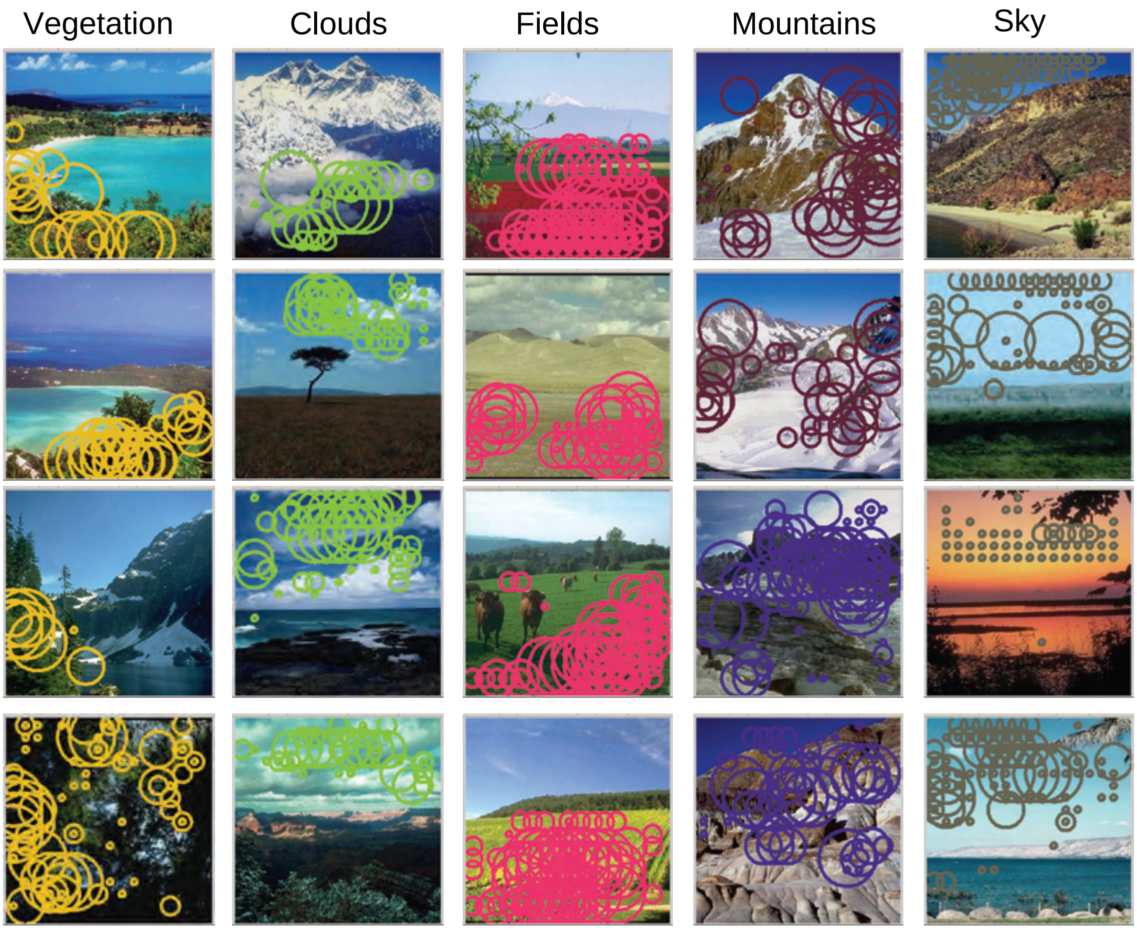


Figure 4.11: Topics segmentation. Five topics (vegetation, clouds, fields, mountains and sky) are shown. Only circular regions with a topic posterior $P(z|w, d)$ greater than 0.8 are shown.

used.

In contrast to previous approaches [41, 102, 150], our topic learning stage is completely unsupervised and we obtain significantly superior performance in the pure bag of words situation (without spatial information). Nevertheless, it is demonstrated in [76] that using the spatial information is very useful for the scene classification task. So that the main issue in next chapter will be to investigate how to incorporate spatial information to our hybrid approach to improve the scene classification performance.

Chapter 5

Adding spatial information

In this chapter we investigate the benefits of introducing spatial information to the problem of scene (coast, forest, kitchen) and object (dolphins, cars, airplanes) classification. In order to introduce spatial information, we were inspired by the method of Lazebnkik et al. [76]. This method is based on repeatedly subdividing the image and computing histograms of local features at increasingly fine resolutions levels obtaining a Pyramid Histogram Of visual Words (PHOW). This descriptor is then used as input for a discriminative classifier. In the first part of the chapter we extend the Spatial Pyramid (SP) framework of Lazebnik et al. [76] in two ways. First we introduce a new Pyramid-radial basis function kernel (P-rbf), suitable for SVM classifier. Second, we generalize the PHOW descriptor, and learn its level weighting parameters (on a validation set). The proposed P-rbf kernel increases performance by 1% *respect to the original pyramid kernel and the weight generalization improves performance another* 1%.

In the second part of the chapter we study how to incorporate spatial information into the hybrid approach proposed in chapter 4. Here we are inspired by the ABS-pLSA framework proposed by Fergus et al. [44]. We use the PHOW which represents the joint density on the appearance and location of each region as input of the generative classifier of the hybrid system. This method is called SP-pLSA, and it improves performance by 2.4% *compared to original SP [76].*

5.1 Introduction

Recently it has been shown [20, 44, 76] that position information can improve scene classification performance (earlier work had shown little benefit [150]). Motivated by this, we consider the problem of image classification where our main goal in this chapter is both, to find an image representation which incorporates spatial information and to explore wether this representation is suitable as input of the generative learning of our hybrid system.

Lazebnik *et al.* [76] successfully incorporated spatial information to the bag-of-words showing very good performances for scene classification. This method works by placing a sequence of increasingly coarser grids over the image and taking a weighted sum of the number of matches that occur at each level of resolution. The image is represented by a Pyramid Histogram Of visual Words (PHOW) descriptor. We extend this method by introducing a new Pyramid radial basis function kernel (P-rbf). This is a kernel which generalizes from the original one and has the advantage that the most suitable similarity distance measure can be used. This kernel is introduced in section 5.2.

The original pyramid level weighting has fixed ratios between the pyramid levels and gives more importance to the matches found at higher resolution levels. However this ratios have not to be the optimum and the higher pyramid level has not to be the best. The PHOW is improved by generalizing the weighting for the levels of the hierarchical spatial histogram [57, 76]. This level-weighting is introduced in section 5.3.

In the second part of the chapter we investigate wether the spatial information can be used within the generative learning. We are inspired by the ABS-pLSA proposed by Fergus *et al.* [44] which incorporates location into the pSLA model by quantizing the location within the image into one of X bins and then to have a joint density on the appearance and location of each region. Though the operation itself seems trivial there is the open question of what is the right subdivision scheme or which is the right number of bins X to use. We can solve this by using the PHOW, and the best results will be achieved when multiple resolutions are combined in a principled way. We use the Spatial Pyramid representation with the optimum level weights as input for the pLSA learning. We call this method as SP-pLSA and it is explained in depth in section 5.4. Figure 5.3 shows an example of our spatial pyramid proposal and how it is used as the input of the generative learning of the hybrid system proposed in previous chapter.

Sections 5.5 and 5.6 describe the datasets, methodology and implementation details. We compare a number of methods that include both latent models and spatial information, and demonstrate improved results over [44, 76] in section 5.7. A comparison with previously published results is provided in section 5.8. A summary and quantization improvement in performance for each of the extensions proposed is given in section 5.9. A discussion about the ambiguities and difficulties of the scene/object classification task is given in section 5.10.

5.2 Pyramid radial basis function kernel – P-rbf

Spatial Pyramid framework - SP

We follow the scheme proposed by Lazebnik *et al.* [76] which is based on spatial pyramid matching [57]. Consider matching two images each consisting of a 2D point set, where we wish to determine soft matches between the point sets when the images are overlaid – for a particular point the strength of the match depends on the distances from its position to points in the other set. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction (like a quadtree). The number of points in each grid cell is then recorded. This is a pyramid representation because the number of points in a cell at one level is simply the sum over those contained in the four cells it is divided into at the next level. The cell counts at each level of resolution are the bin counts for the histogram representing that level. The soft correspondence between the two point sets can then be computed as a weighted sum over the histogram intersections at each level. Similarly, the lack of correspondence between the point sets can be measured as a weighted sum over histogram differences at each level.

In the image case, the pyramid matching is applied to the two-dimensional image space, and a BoW vector is computed for each grid cell at each pyramid resolution level. So the 2D points in the example above are replaced by visual words obtaining a Pyramid Histogram Of visual Words (PHOW) descriptor for the image. The kernel (used within and SVM classifier) is defined as follow:

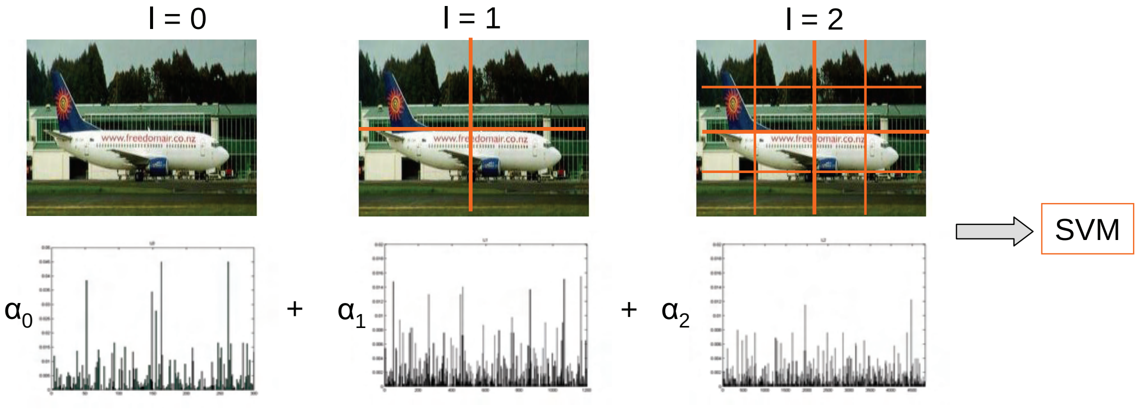


Figure 5.1: Spatial Pyramid [76] framework. The image is recursively split, and a BoW vector is then computed for each grid cell at each pyramid resolution level.

$$K^L(X,Y) = \sum_{v=1}^{V} k^L(X_v, Y_v) \tag{5.1}$$

where V is the vocabulary size and $k^L(X,Y) = \Sigma_{l=0}^{L} \alpha_l I^l$, and I is the histogram intersection between the two feature vectors X and Y at each pyramid level. Normally matches found at finer resolutions are weighted more highly than matches found at coarser resolutions. The PHOW is normalized to sum the unity. A graphical example of this method is shown in figure 5.1.

In forming the pyramid the grid at level l has 2^l cells along each dimension. Consequently, level 0 is represented by a V-vector corresponding to the V visual words of the histogram, level 1 by a $4V$-vector etc, and the PHOW descriptor of the entire image is a vector with dimensionality $V \sum_{l \epsilon L} 4^l$. For example, for levels up to $L = 1$ and $K = 200$ bins it will be a 1000-vector.

P-rbf kernel

The kernel above uses the histogram intersection distance to measure the similarity between two descriptors at each level. However this distance has not to be the best one to find the similarity between a couple of descriptors. We generalize the kernel by introducing a Pyramid-radial basis function kernel:

$$K(D_I, D_J) = exp\{\frac{1}{\beta}\sum_{l=0}^{L}\alpha_l d_l(D_I, D_J)\} \tag{5.2}$$

where d_l is the distance between D_I and D_J at pyramid level l. We use the χ^2 on the normalized PHOW descriptors to compute it, as it is demonstrated to be a good distance for histogram comparison [158]. It is shown that this kernel increases performance compared to kernel in (5.1) in section 5.7. Implementation details are given in section 5.5.

5.3 Global Level-Weights – GLW

In the original spatial pyramid representation [76] each level was weighted using $\alpha_l = 1/2^{(L-l)}$ where L is the number of levels and l the current level. This means that histograms from finer resolutions are weighted more highly than those at coarser resolutions. However, this may not be the optimum weight choice. We investigate a method to optimally learn the weights called Global Level-Weights (GLW).

Instead of giving a fixed weight to each pyramid level as in [76], we *learn* the weights α_l which give the best classification performance over all categories on the validation set (see section 4.5). Consequently, the finer resolutions may not be given the highest weight. In this case, the number of parameters to *learn* is the same as the number of pyramid levels we explore. For example if we explore up to $L = 2$ we need to *learn* 3 parameters α_0, α_1 and α_2. Implementation details are given in section 5.7.

5.4 Spatial Pyramid-pLSA – SP-pLSA

ABS-pLSA

This is the method proposed in [44] and was applied for object recognition. The pLSA model is extended to incorporate location information by quantizing the location within the image into one of X bins. The joint density on the appearance and location of each region is then represented. Thus $P(w|z)$ in pLSA becomes $P(w, x|z)$, a discrete density of size $(W \times X) \times Z$. The same pLSA update equations outlined in section 4.2 can be

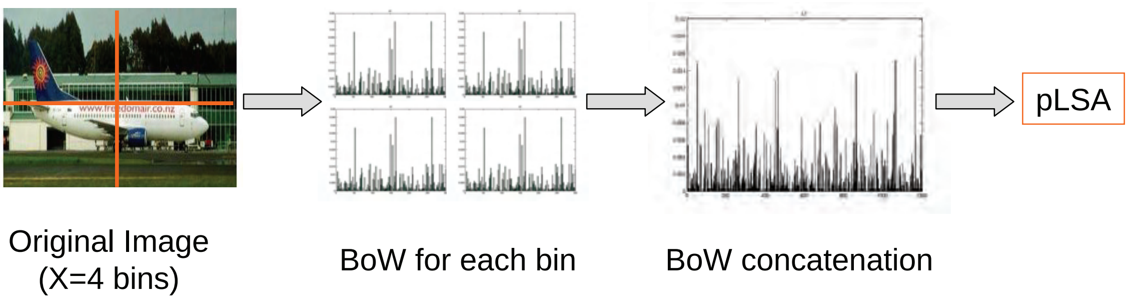


Figure 5.2: ABS-pLSA [44] framework. Spatial information is incorporated into pLSA by quantizing the location into each bin.

easily applied to this model in learning and recognition. The case $X = 1$ corresponds to standard pLSA with no spatial information. The size of the feature vector is XV, (X times the vocabulary size V). A graphical example of this method is shown in figure 5.2.

SP-pLSA

Our proposal is inspired by both previous ones, ABS-pLSA and SP. We incorporated location information in pLSA by using the X bins at each resolution level L, weighting the bins for each level (α_l) as in SP. Note that in ABS-pLSA only the bins for one resolution level are used and in SP-pLSA we use the weighted bins for L resolutions. So for example when $L = 1$, if using ABS-pLSA we have $X = 4$ bins, and if we use SP-pLSA we have $X = 5$ bins (one bin for $L = 0$ and four bins for $L = 1$). Thus $P(w|z)$ in pLSA becomes $P(w, x, l|z)$. The same pLSA update equations outlined in section 4.2 can be easily applied to this model in learning and recognition. The size of the PHOW descriptor is $V \sum_{l \epsilon L} 4^l$ as in SP. The PHOW is normalized to sum to unity. A graphical example is shown in figure 5.3.

5.5 Datasets and methodology

Datasets

We evaluated the improvements introduced in this chapter for scene classification on the four scene datasets (OT, VS, FP and LSP) and for the object classification on Caltech-101.

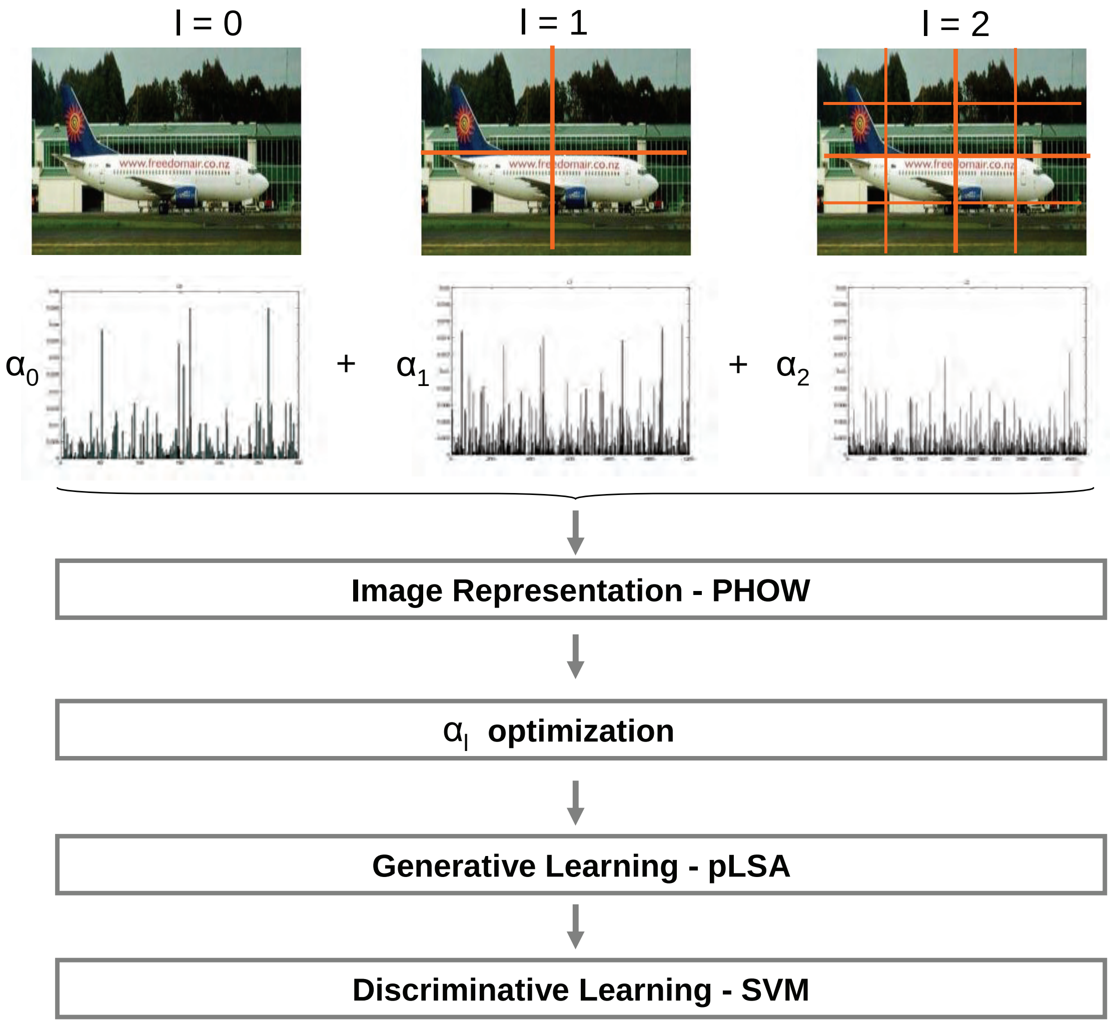

l = 0
l = 1
l = 2
α0
+
α1
+
α2
Image Representation - PHOW
αl optimization
Generative Learning - pLSA
Discriminative Learning - SVM

Figure 5.3: Generative/Discriminative hybrid system using spatial information.

See chapter 3 for a complete description about them.

Methodology

The performance is measured using a confusion table, and overall performance rates are measured by the average value of the diagonal entries of the confusion table. This process is repeated 10 times and the mean average and standard deviation are given.

For the SP framework we use an M-SVM classifier with the kernel in (5.2) and PHOW. β is the average of $\sum_{l=0}^{L} \alpha_l d_l(D_I, D_J)$ over the training data, α_l is the optimized weight at level l (see section 5.7.1). For the hybrid system SP-pLSA we use the generative learning with the PHOW descriptor and then the $P(z|d)$ is used as input for the M-SVM discriminative classifier, with an L_2 kernel. ABS-pLSA is evaluated for X = 1, 4 and 16 bins.

Moreover we implemented a baseline method for comparison:

xy-pLSA. The x and y normalized position of each pixel is concatenated to the feature vector. So in this case the dimension of the feature vector is $N^2 \times 3 + 2$. Each component of the feature vector (both spatial and SIFT) is in the range $[0, 1]$. However, the SIFT part of the vector is sparse in general.

5.6 Implementation

Scene datasets

For each dataset we use SIFT and four circular supports spaced at $M = 10$; the parameters V,Z,α_0, α_1 and α_2 have the optimized values for each dataset (see table 4.5 and table 5.1). For the PHOW descriptor we only explored up to $L = 2$ which was demonstrated in [76] to be the optimum level. We used colour for OT and VS, and grey for FP and LSP. The visual vocabulary is computed independently for each dataset, as described in section 4.5.

Caltech-101

For the experiments, four concentric circles SIFT with colour information are used to represent each pixel, spaced at $M = 10$, $V = 300$, $Z = 80$ topics for SP-pLSA. The weights ratios for GLW are $\alpha_0 : \alpha_2 = 0.9$ and $\alpha_1 : \alpha_2 = 0.8$ for SP-pLSA, and $\alpha_0 : \alpha_2 = 1$ and $\alpha_1 : \alpha_2 = 0.9$ for SP. For the PHOW descriptor we only explored up to $L = 3$ to prevent object overfitting. We carried out experiments using 15 and 30 random training images per category, and 50 random testing images per class (disjoint from the training images). The mean recognition rate per class is used so that more populous (and easier) classes are not favored.

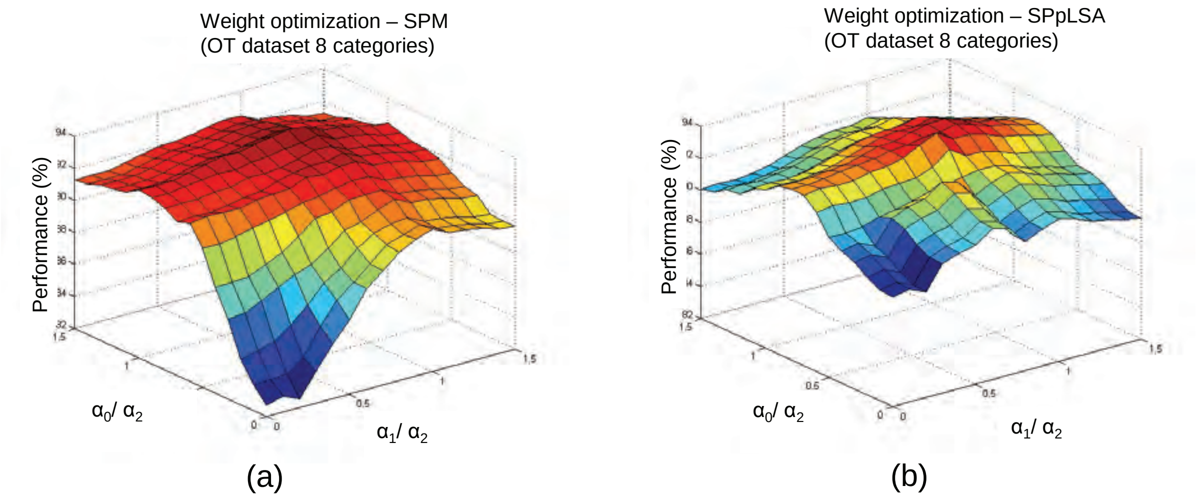


Figure 5.4: Optimization rates between the weights at each pyramid level using the validation set from the OT dataset: (a) SP is used; (b) SP-pLSA is used.

ratios	OT - 8	OT - $4N$	OT - $4MM$	VS	FP	LSP
			SP weights			
$\alpha_0 : \alpha_2$	1	0.8	0.9	1	0.9	1
$\alpha_1 : \alpha_2$	0.9	0.9	0.8	0.9	0.8	0.8
			SP-pLSA weights			
$\alpha_0 : \alpha_2$	0.7	0.9	0.9	0.9	1	1
$\alpha_1 : \alpha_2$	0.8	0.8	0.8	0.7	0.9	0.8

Table 5.1: Optimized weight ratios $\alpha_0 : \alpha_2$ and $\alpha_1 : \alpha_2$ for each dataset using the validation set. $4N$ = 4 Natural categories; $4MM$ = 4 Man-Made categories.

5.7 Measuring the improvements

5.7.1 Optimizing the parameters α_0, α_1 and α_2 (on the validation set)

GLW. Since the final PHOW is normalized there are only two independent parameters which represent three of the ratios in: $\alpha_0 : \alpha_1 : \alpha_2$. Using the validation set we optimize the ratio between the weights α_0:α_2 and α_1:α_2 over the range $[0, 1.5]$. Figure 5.4a shows the performance when optimizing using SP on the OT dataset (8 categories). Figure 5.4b shows performances when optimizing the weights for SP-pLSA on the same dataset. The optimized ratios using the validation set are summarized in table 5.1.

L	pLSA	xy-pLSA	ABS-pLSA	SP (5.1)	SP (5.2)	SP-pLSA
L = 0	$87.1(\pm 0.9)$	$89.0(\pm 0.6)$	$87.1(\pm 0.8)(X = 1)$	$82.6(\pm 0.7)$	$83.8(\pm 0.7)$	$87.1(\pm 1.0)$
L = 1	–	–	$87.9(\pm 0.9)(X = 4)$	$89.4(\pm 0.6)$	$90.3(\pm 0.7)$	$90.7(\pm 0.8)$
L = 2	–	–	$88.3(\pm 0.9)(X = 16)$	$90.2(\pm 0.7)$	$91.0(\pm 0.6)$	**91.1**(± 0.9)

Table 5.2: Performance comparison for the OT dataset when spatial information is used. Four concentric circles spaced at $M = 10$ and $V = 1500$, $Z = 25$, $\alpha_0 : \alpha_2 = 0.5$ and $\alpha_1 : \alpha_2 = 0.5$ (the same weight ratios used in [76]).

5.7.2 Spatial frameworks comparison

In this section we compare the two reviewed methods (ABS-pLSA and SP framework) with the hybrid method without spatial information, the SP-pLSA and the baseline xy-pLSA. The OT dataset (8 categories) is used.

Table 5.2 shows the values for the hybrid method without position and the four methods above described. For the SP results with both kernels (5.1),(5.2) are included. The weights used in this experiments are: $\alpha_0 = 0.25$, $\alpha_1 = 0.25$ and $\alpha_2 = 0.5$ (the same weights are used in [76]). When only the first level of the pyramid is used ($L = 0$) the best result (89.0%) is obtained when using xy-pLSA. In this case SP works directly over the BoW and has worse results than the methods that use pLSA. When $L = 1$ and $L = 2$ the best results are obtained for SP-PLSA (90.7% and 91.1%) followed by SP with the proposed kernel (90.3% and 91.0%). The proposed radial basis kernel (6th column) increases performance by 1% respect to the original pyramid kernel (5th column). Unless stated otherwise we are going to use (5.2) with the SP framework.

When using GLW optimization, for SP the best performance (92.2% for the test data) is for $\alpha_0 : \alpha_2 = 1$ and $\alpha_1 : \alpha_2 = 0.9$. For the SP-pLSA framework the performance increases to 92.7% for the test data using the validation set optimized ratios $\alpha_0 : \alpha_2 = 0.7$ and $\alpha_1 : \alpha_2 = 0.8$. Note that best performances are obtained for higher ratios and they exceed that given in table 5.2 by 1%. Two main conclusions can be extracted from these results: (i) higher weight is given to finer levels, so they have more importance in order to represent the image, but even though (ii) a significative weight is also given to all the other lower levels, so that they all are significative to describe the image.

Dataset	# of categ.	# train	# test	pLSA	SP	SP-pLSA	Authors
OT	8	800	1888	83.5 ± 1.2	87.1 ± 0.8	**87.8** ± 1.4	83.7 [102]
OT	4 N	1000	472	90.7 ± 0.9	93.3 ± 0.7	**93.9** ± 1.2	89.0 [102]
OT	4 MM	1000	216	91.7 ± 0.8	94.2 ± 0.9	**94.8** ± 1.3	89.0 [102]
VS	6	600	100	87.8 ± 1.0	**88.6** ± 0.9	88.3 ± 1.4	74.1 [150]
FP	13	1300	2459	74.3 ± 1.3	85.5 ± 0.8	**85.9** ± 1.3	65.2 [41]
LSP	15	1500	2986	72.7 ± 1.2	83.5 ± 0.8	**83.7** ± 1.3	81.4 [76]

Table 5.3: Comparison of our algorithm with other methods using their own databases. $L = 2$ for SP-pLSA and SP. GLW optimization. L_2 kernel is used for the discriminative classifier of the hybrid system. Kernel in (5.2) is used for SP.

Summary

We outperform the (SP) classifier proposed in [76] by 1% when working with the proposed kernel. Moreover if we use the GLW optimization we increase performance another 1%, making a total of 2% respect the original SP framework. We successfully incorporated spatial information into the pLSA framework (SP-pLSA) obtaining slightly better performances (0.2%) than SP when using level-weights optimization and the P-rbf kernel. Performance increases by 4%, compared to results shown in chapter 4, if spatial information is used.

5.8 Comparison to previous results

5.8.1 Scene classification

We compare the performance of our scene classification algorithm to the supervised approaches of Vogel and Schiele [150] and Oliva and Torralba [102], and the semi-supervised approach of Fei-Fei and Perona [41] and Lazebnik et *al.* [76], using the same datasets (VS,OT,FP and LSP respectively – see chapter 3) that they tested their approaches on and the same number of training and testing images.

The results are given in table 5.3. Without using spatial information (5th column in table 5.3) our method outperformed the previous methods in [41, 102, 150]. Better results

# train	[76]	[11]	[58]	[100]	[151]	[157]	SP	SP-pLSA
15	56.4	52.0	49.5	51.9	44.0	59.0	$58.7(\pm 0.8)$	**59.3**(± 1.4)
30	64.6	–	58.2	56.0	63.0	66.0	$66.5(\pm 0.7)$	**67.1**(± 1.5)

Table 5.4: Classification of Caltech 101 with 15 or 30 training images per class. For SP-pLSA and SP four concentric circles spaced at $M = 10$ are used, $V = 1500$, $Z = 80$, and SVM is used as the discriminative classifier.

are obtained with spatial information (6th and 7th columns). We have better performances (83.7%) than in [76] when using SP-pLSA and also when using their own method with our features, kernel and level-weight optimization. In [76] with $V = 400$ words and weight ratios $\alpha_0 : \alpha_2 = 0.5$, $\alpha_0 : \alpha_1 = 0.5$ they achieve 81.4% of correct classified scenes. Using SP with theses same parameters and the P-rbf kernel (5.2) our performance increases up to 82.5%. In both our and their experiments grey SIFT descriptors are used. This demonstrates again that the P-rbf kernel increases performance.

5.8.2 Object classification – Caltech-101

Caltech-101 contains 101 object categories to distinguish amongst. Chapter 3 provides a complete description of this dataset. [76] argued that Caltech-101 was essentially a scene matching problem so an image based representation was suitable. Their representation added the idea of flexible scene *correspondence* to the bag-of-visual-word representations that have recently been used for image classification [35, 112, 158]. So that this dataset is also useful to test our scene classification algorithm.

A number of previously published papers have reported results on this data set: Lazebnik et *al*. [76], Berg et *al*. [11], Grauman and Darrell [58], Zhang et *al*. [157] etc.

Table 5.4 shows our results and those reported by other authors. Our best performance is when using SP-pLSA algorithm with a mean recognition rate of 59.3% with 15 training images per class, and 67.1% with 30 training images per class. This outperforms the results reported by Zhang et *al*. [157] that to our knowledge were the best until now.

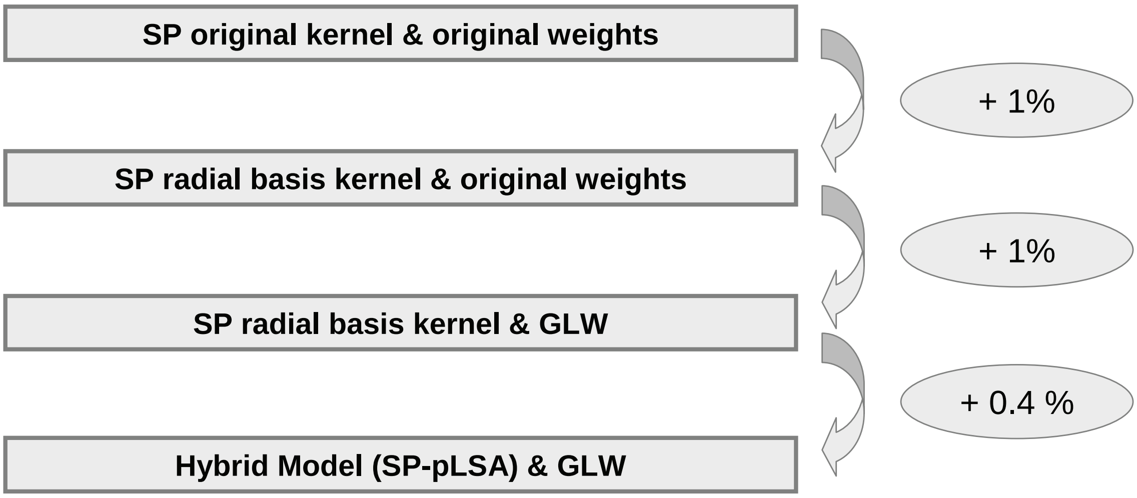

SP original kernel & original weights
SP radial basis kernel & original weights
SP radial basis kernel & GLW
Hybrid Model (SP-pLSA) & GLW
+ 1%
+ 1%
+ 0.4 %

Figure 5.5: This graph shows how much each representational and feature increment contributes to the overall performance.

5.9 Summary

Following the approach in [76], a spatial grid at different resolution levels is used to represent the images. The novelty arises in: (i) the kernel; (ii) the level-weights. We introduced a new P-rbf kernel which increases performance by 1%. We demonstrated that globally optimizing the weights increases performance another 1%. We successfully incorporate this spatial pyramid descriptor within pLSA (SP-pLSA). In this case performance slightly increases by 0.4%. Figure 5.5 attributes how much each representational and feature increment contributes to the overall performance.

5.10 Discussion

5.10.1 The scene classification task

We discuss here the results obtained with and without spatial position information when using the hybrid system for scene classification.

Figure 5.6a shows the confusion matrix between the 8 categories in OT dataset when no spatial information is used. The best classified scenes are *highway* and *forest* with a performance of 89.8% and 98.8% respectively. The most difficult scenes to classify are

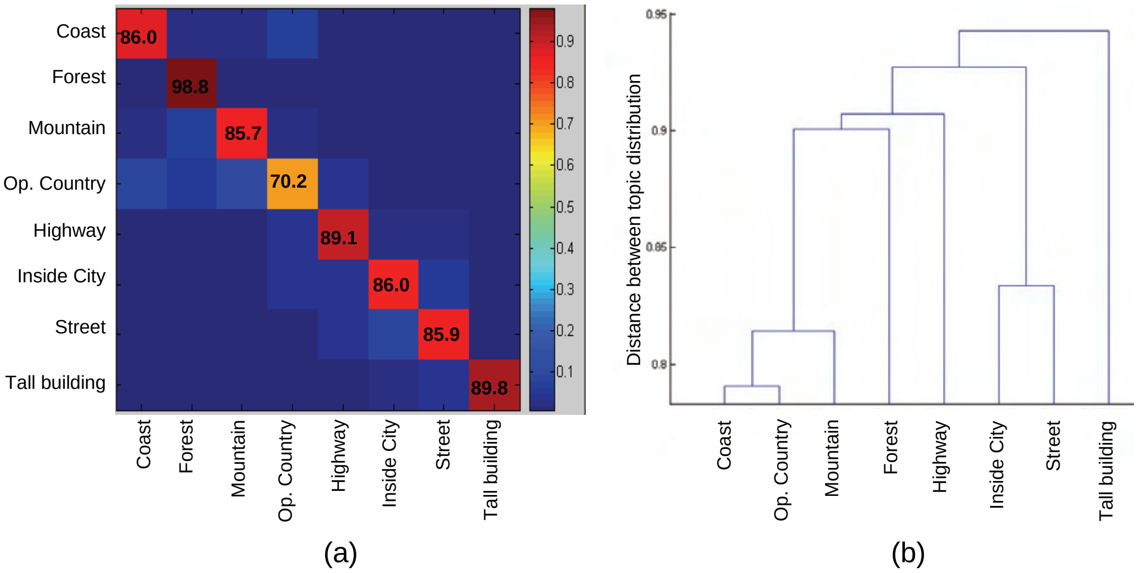


Figure 5.6: (a) Confusion matrix for the 8 categories in the OT dataset. (b) Dendrogram showing the closest categories, which are also the most confused.

open country. There is confusion between the *open country* and *coast* scenes, and between the *open country* and *mountain scenes*. The most confused man made images are *street*, *inside city* and *highway*. These are also the most confused categories in [102]. We can also establish some relationship amongst the categories by looking at the distances among the topic distributions between them (see the dendrogram in figure 5.6b). When the topic distributions are close, the categories are also close to each other on the dendrogram. For example, the closest natural categories are *open country* and *coast* and the closest man-made are *inside city* and *street*.

Figure 5.7 shows some confused images between categories showing the ambiguity between some of them. Scene categorization is characterized by potential ambiguities since it depends strongly on the subjective perception of the viewer. For example some of the *open country* images shown in figure 5.7a can be easily classified as *mountain* for some humans as the system did. Obviously, the obtainable classification accuracies depend strongly on the consistency and accuracy of the manual annotations, and sometimes annotation ambiguities are unavoidable. For example, the annotation of *mountains* and *open country* is quite challenging. Imagine an image with *fields* and *snow hills* in the far distance: is it *open country* or *mountain*? Even more confused are *coast* and *open country* scenes (figure 5.7b) yet both of them have a similar structure: *water* or *fields* and the *sky*

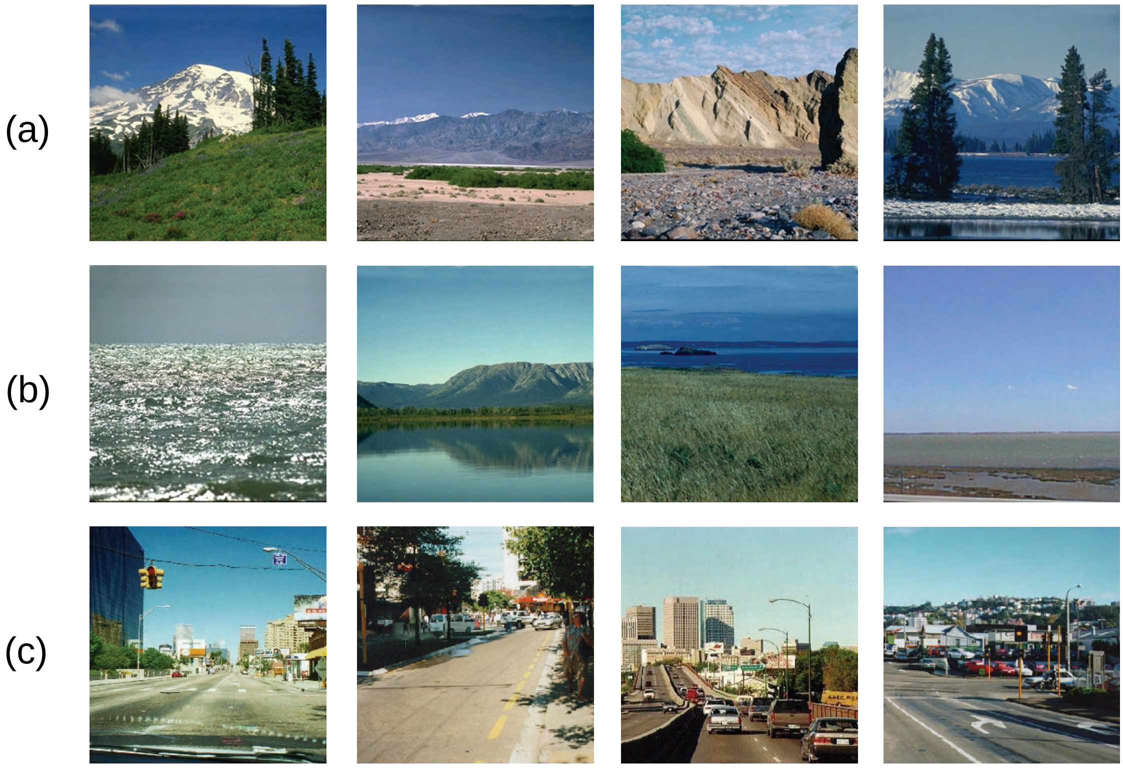


Figure 5.7: Images showing the most confused categories: (a) open country images confused as mountains; (b) coast images confused as open country; (c) highway images confused as street.

in the distance. For that reason, it is not surprising that *coast* and *open country* are confused in both directions. Another major confusion appears between *streets* and *highway*. This results mainly from the fact that each street scene contains a *road* whereas the most important part of highway scenes is the *road*. *Streets* and *inside city* images are confused because normally streets occur in cities.

Let's see what happens when spatial information is included. Figure 5.8a shows the confusion matrix for the 8 categories in the OT dataset when using SP-pLSA with $L = 2$. Now for the *forest* scenes we obtain a rate of 100% of correct classified images, and all the classification rates for the other scenes are also increased. Again the most difficult scenes to classify are the *open country*. Figure 5.8b shows some images well classified using SP-pLSA and wrongly classified without spatial information. This demonstrates that spatial distribution can reduce the ambiguity – or at least that spatial distribution correlates with the annotator's choices. However we are still far from 100% correct classification, again

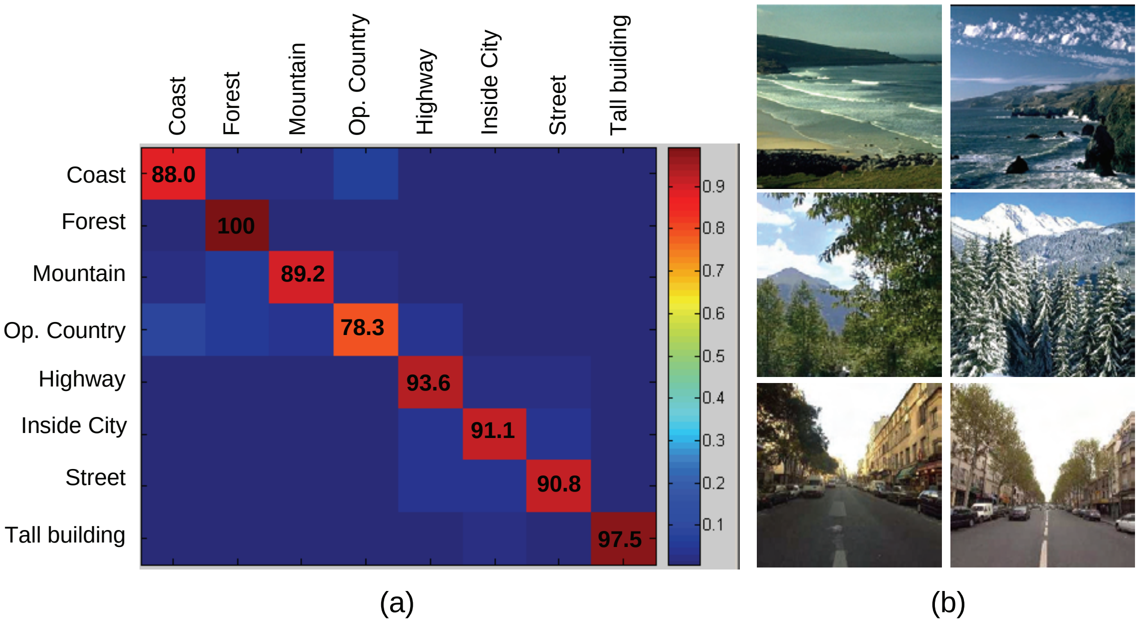
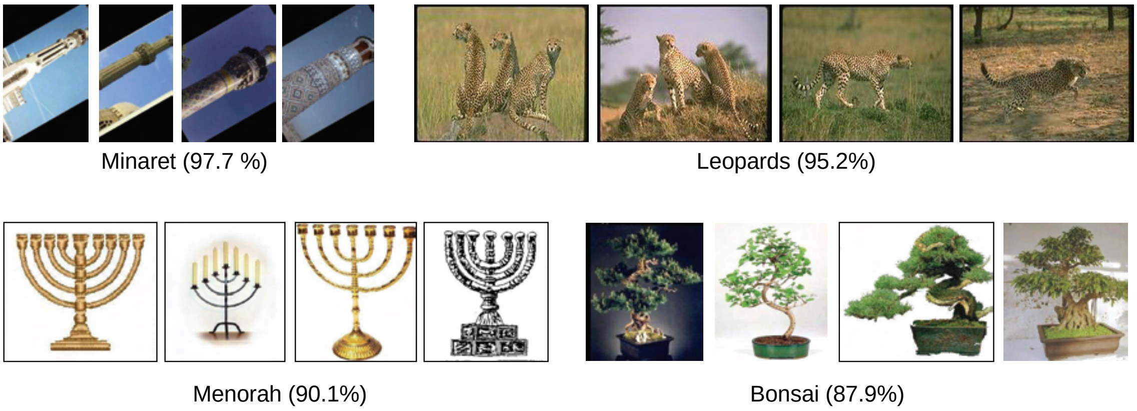


Figure 5.8: (a) Confusion matrix for the 8 categories in the OT dataset when using SP-pLSA. (b) Top: two coast scenes confused as mountain when spatial information is not used, and well classified using SP-pLSA; Middle: two forest scenes confused as mountains without spatial information, and well classified with SP-pLSA; Bottom: two street scenes confused as highway without spatial information, and well classified using SP-pLSA.

due to the ambiguities between the scene categories used. Vogel and Schiele [150] analyzed in detail the ambiguities between scene categories, showing that there is a semantic transition between categories. Their experiments with human subjects showed that many images cannot be clearly assigned to one category. How far away must a *mountain* be so that the image moves from the *mountains* category to the *open country* category? How much *road* is necessary to make a *street* image into a *highway* image and vice versa? And we arrive at the same conclusion as [150]: it is not wise to aim for a hard decision categorization of scenes. However, since scenes, that is full images, contain very complex semantic details, hard scene categorization is an appropriate task for: (i) testing the image representation [150], in this case provided by topics, (ii) having an approximation on how the ranking on an image retrieval system would work, and (iii) classifying mutually exclusive scenes such as indoor/outdoor, garden/bathroom or coast/kitchen.

We have done some preliminary experiments with k-means clustering the image topics



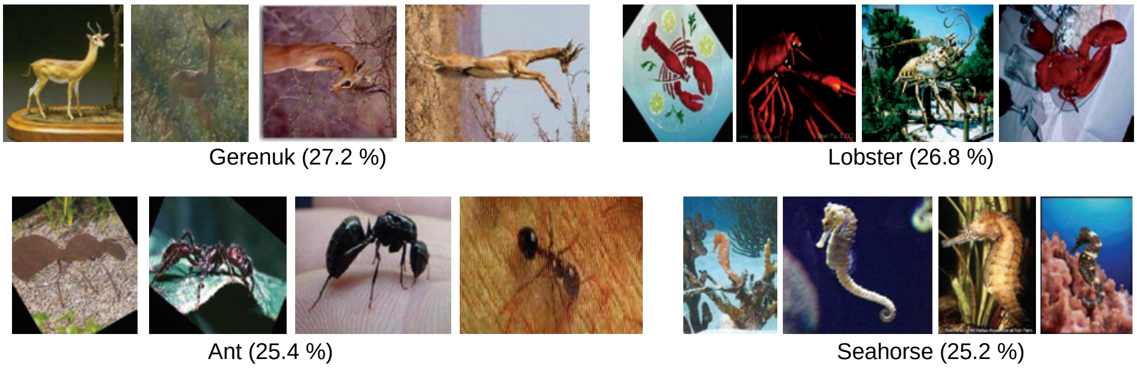
Minaret (97.7 %)

Leopards (95.2%)

Menorah (90.1%)

Bonsai (87.9%)

Figure 5.9: Some of well classified images on Caltech-101 for SP-pLSA. The percentage is the correct classification rate for each class.



Gerenuk (27.2 %)

Lobster (26.8 %)

Ant (25.4 %)

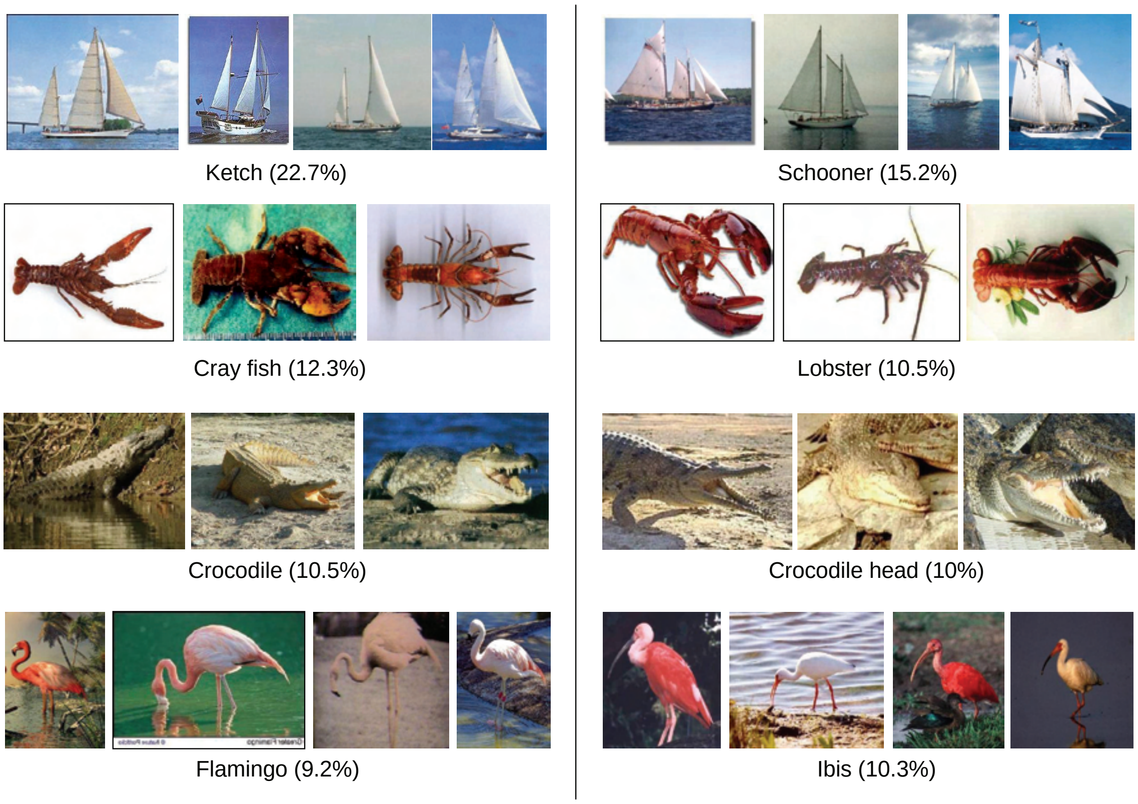
Seahorse (25.2 %)

Figure 5.10: Some of not so well classified images on Caltech-101 for SP-pLSA. The percentage is the correct classification rate for each class.

provided by SP-pLSA to automatically detect visually similar categories. The results are interesting because the resulting clusters had a semantic meaning such as fields with mountains at the back, fields with flowers, coasts with rocks, sunshine coast, highway with cars and without cars etc. Nevertheless, the images with semantic transition between categories are not well clustered (because there are not sufficient ambiguous images). A solution would be to use EM soft assignment in the clustering.

5.10.2 The object classification task

Figure 5.9 shows a few of the object classes easiest to classify for SP-pLSA. As in [76], the successful classes are either dominated by rotation artifacts (like *minaret*), have very little



Ketch (22.7%)

Schooner (15.2%)

Cray fish (12.3%)

Lobster (10.5%)

Crocodile (10.5%)

Crocodile head (10%)

Flamingo (9.2%)

Ibis (10.3%)

Figure 5.11: Most confused images on Caltech-101 for Sp-pLSA. Images on the left are images confused as class on the right and images on the right are images confused as class on the left. The percentage is the miss-classification rate.

clutter (like *menorah*), or represent coherent natural scenes (like *leopards* or *bonsai*). Note that all theses classes have very similar colors for both the objects and the background. On the other hand, figure 5.10 shows a few of the object classes hardest to classify for our method. The least successful classes are those with more intra-colour variation and those where the objects have more translation variation. These two features (colour and translation) can be appreciated in the categories *gerenuk*, *seahorse*, *ant* and *lobster* of figure 5.10.

Figure 5.11 shows the top five of the most confused categories. Images ont he left show confused images with the category in the right, and images on the right are confused image with the category on the left. The percentage is the miss-classification rate. For example, in the first row the four first *ketch* images are confused as *schooner* with a miss-classification rate of 22.7%, and the four *schooner* images are confused as *ketch* with a

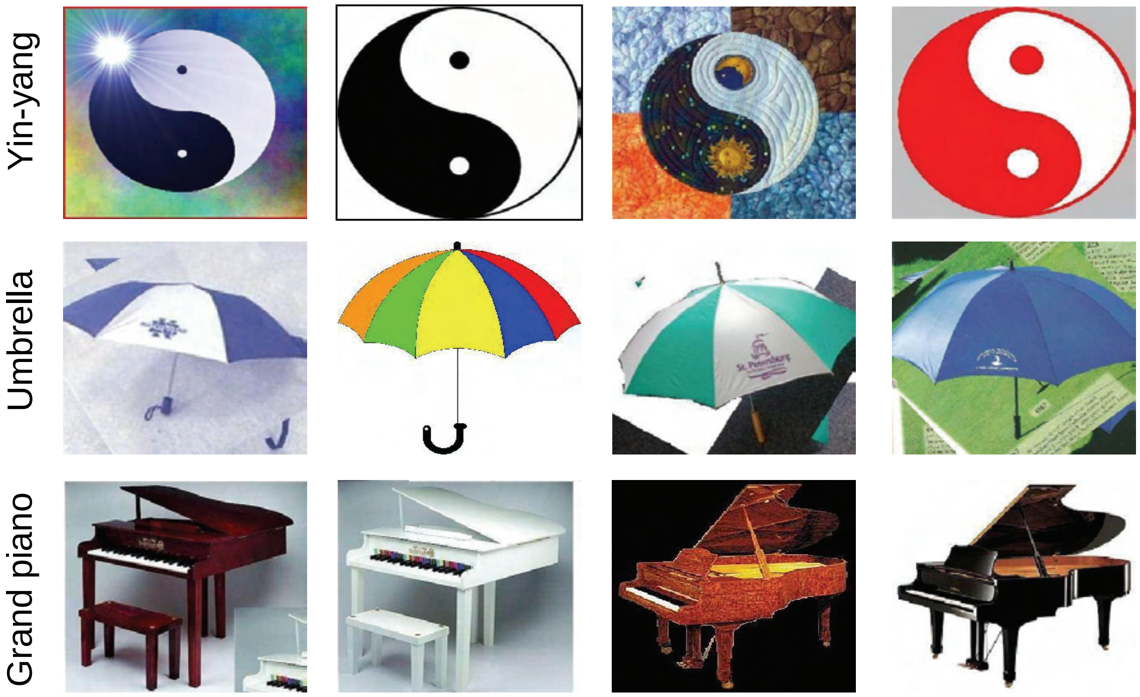


Figure 5.12: Appearance or shape features? these images from Caltech-101 show us that there are some categories which can be better recognized using its shape.

miss-classification rate of 15.2%. All of them are between closely related classes and have a similar colour: *cray fish* is confused by *lobster* and viceversa, *crocodile* is confused by *crocodile head*, *flamingo* is confused by *ibis* etc.

Most of the confusions are due to the colour of objects and background. Colour features are good for scenes so most of them can easily be recognized by its appearance (mainly outdoor scenes). However if we have a look to the Caltech-101 categories we can clearly see that for most of the categories colour does not help. In figure 5.12 we can see that the *yin-yang* category or the *umbrella* category could be easily recognized by its shape information. This clearly shows that colour is not discriminative to distinguish between them. However, we can not forget that there are also a few categories (e.g. *leopards* and *ketch* in figure 5.13) for which colour is an important feature.

In next chapter we introduce a new shape descriptor suitable for object classification, and we propose a new method for merging features. The translation variance amongst objects will be addressed in chapter 7.

An observation about computational cost needs to be done at this point. For the

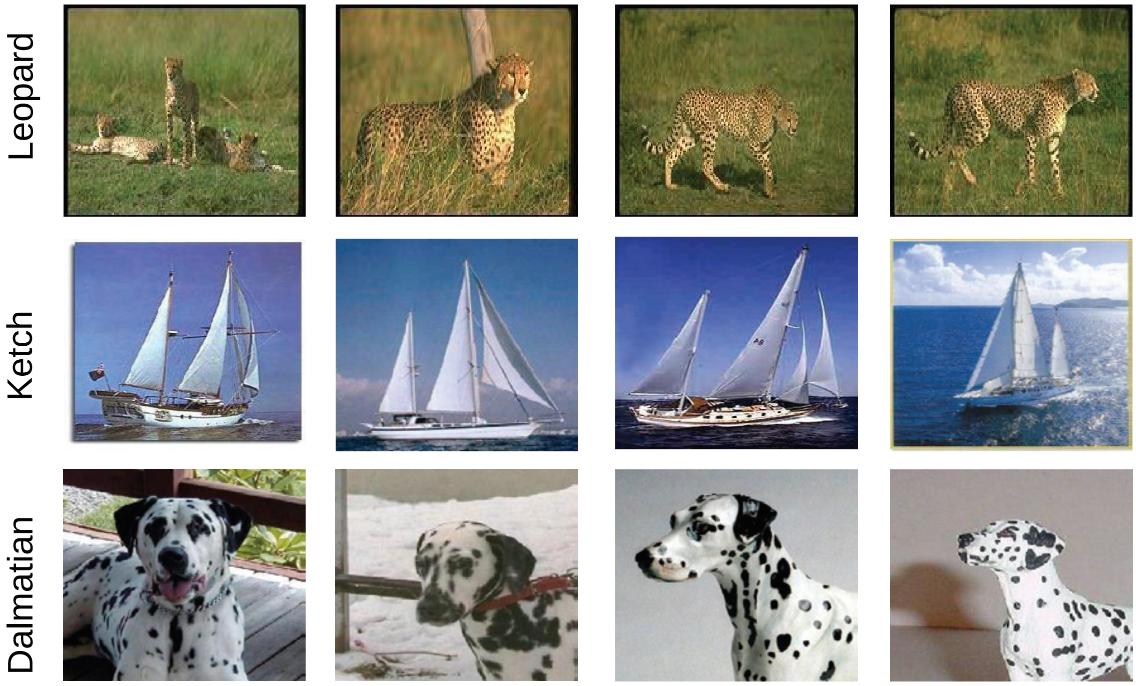


Figure 5.13: Appearance or shape features? these images from Caltech-101 show us that there are some categories which can be better recognized using its appearance.

Caltech-101 dataset, the time to fit the SP-pLSA model when using the hybrid system is 46 hours approximately (Matlab implementation on a 1.7 GHz computer). This is very expensive due to the high-dimensionality of the descriptor and the number of categories. Hence we need to think about a trade-off between performance and computational cost. Since the SP-pLSA only increases performance by 0.4% it is useful if we work with few categories (eg. from 4 to 15) however when working with the 100 categories in Caltech-101 it is very expensive to use the hybrid system and comparable results can be obtained when using the PHOW representation with the P-rbf kernel. In the following chapters we will use this last method (SP with P-rbf) for efficiency.

Chapter 6

A pyramid shape descriptor and merging features

The objective of this chapter is to introduce a shape descriptor and to study how different descriptors can be merged for image classification. There are three areas of novelty. First, we introduce a descriptor that represents local image shape and its spatial layout, together with a spatial pyramid kernel. These are designed so that the shape correspondence between two images can be measured by the distance between their descriptors using the kernel. Second, we generalize the spatial pyramid kernel, and learn its class-specific level weighting parameters (on a validation set). Third, we show that shape and appearance kernels may be combined (again by learning parameters on a validation set).

Results are reported for classification on Caltech-101 *and Caltech-*256 *and retrieval on the TRECVID 2006 data sets. For Caltech-*101 *it is shown that the class specific optimization that we introduce exceeds the state of the art performance by more than* 10%*. For Caltech-*256 *we outperform the state of the art by* 5%

6.1 Introduction

We consider the problem of image classification where our main goal is to explore how the spatial distribution of shape can benefit recognition. Much recent work has used a “bag of (visual) words” representation together with an SVM classifier in order to classify images by the objects they contain [35, 158]. These methods represent local appearance patches, but not shape directly. However, representations of shape using the spatial distribution of edges such as [108, 126] often perform as well as or better than local appearance patches, but recognition involves Hough like accumulators in order to assess geometrical consistency, losing the simple vector representation of a bag of words. We introduce a new descriptor which has the advantages of both: it captures the spatial distribution of edges, but is formulated as a vector representation. Similarity on descriptor vectors between two images (for example measured using histogram intersection or χ^2) then measures the similarity of their spatial distribution of edges.

Our descriptor is mainly inspired by two sources: (i) the use of the pyramid representation of Lazebnik *et al.* [76] and (ii) the Histogram of Orientation Gradients (HOG) of Dalal and Triggs [36].

In essence, we wish to assess how well an exemplar image matches (the shape of) another image. As in [76] the intuition is that a strong match goes beyond a “bag of words” and also involves a spatial correspondence. To this end we extend the method of [76] in two ways: the first is to represent shape in the form of edges, replacing the use of quantized appearance patches (visual words). The second extension is to learn a class specific weighting for the levels of the hierarchical spatial histogram [57, 76]. This captures the intuition that some classes are very geometrically constrained (such as a stop sign) whilst others have greater geometric variability (e.g. dolphins, boats). The details of the descriptor, termed PHOG (for Pyramid of Histograms of Orientation Gradients) are given in section 6.2, and the idea is illustrated in figure 6.1. We compare the PHOG descriptor to a standard shape descriptor, Chamfer matching, in section 6.2.3.

The flexibility of the spatial histogram level weighting means that a spectrum of spatial correspondences between two images can be represented. If only the coarsest level is used, then the descriptor reduces to a global edge or orientation histogram, such as used by [66, 138]. If only the finest level is used, then the descriptor enforces correspondence for tiles (spatial bins) over the image. This extreme is what is captured by [36, 134] where histograms are computed over local image regions. Other weightings of the spatial levels capture geometric consistency between these extremes.

Having developed the PHOG descriptor we then introduce kernels, suitable for an SVM classifier, that combine both appearance (visual words) and edge (PHOG) descriptors. This is a form of feature combination and selection, but here the selection is at the kernel level. Again, in a class-specific learning step, the descriptors (appearance or shape or both) most suitable for a particular class are determined. For example, a category such as *car* is best described by shape alone, *leopard* by appearance alone, and *buddha* by a combination of the two. The kernels are described in section 6.3.

Sections 6.4–6.6 describe the datasets used, implementation details, and the experimental procedure and results on classification for Caltech-101 and Caltech-256, and retrieval for TRECVID 2006. It is shown that the set of innovations introduced here lead to a 10% performance improvement over the previous best result on Caltech-101 [157] and a 5% improvement over Caltech-256 [59].

6.2 Spatial shape descriptor – PHOG

Our objective is to represent an image by its local shape *and* the spatial layout of the shape. Here local shape is captured by the distribution over edge orientations within a region, and spatial layout by tiling the image into regions at multiple resolutions. The idea is illustrated in figure 6.1. The descriptor consists of a histogram of orientation gradients over each image subregion at each resolution level – a Pyramid of Histograms of Orientation Gradients (**PHOG**). The distance between two PHOG image descriptors then reflects the extent to which the images contain similar shapes *and* the extent to which the shapes correspond in their spatial layout.

The following sub-sections describe these two aspects (local shape and spatial layout correspondence) in more detail.

6.2.1 Local shape

Local shape is represented by a histogram of edge orientations within an image subregion quantized into K bins. The contribution of each edge is weighted according to its magnitude, with a soft assignment to neighboring bins in a manner similar to SIFT [84]. Implementation details are given in section 6.5.

Each bin in the histogram represents the number of edges that have orientations within

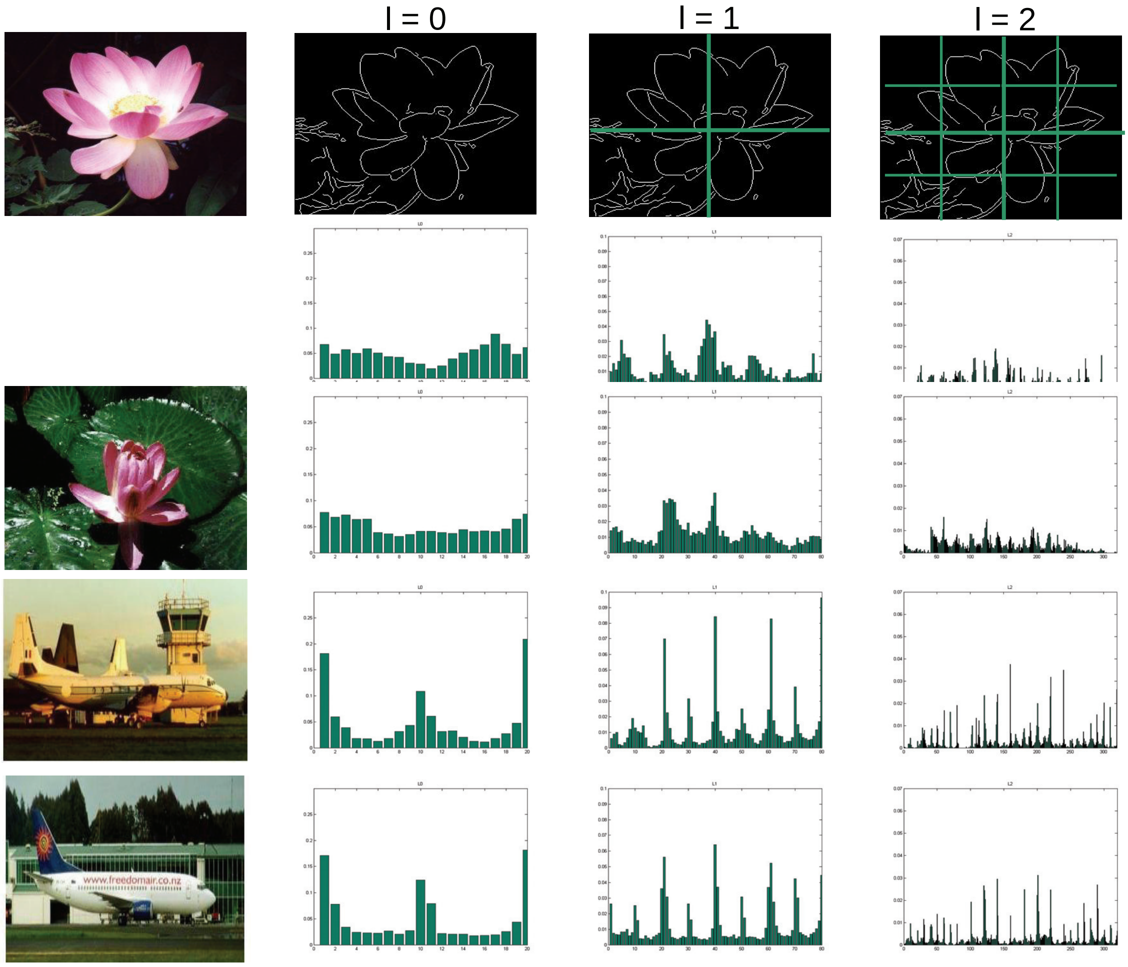


Figure 6.1: Shape spatial pyramid representation. Top row: an image and grids for levels $l = 0$ to $l = 2$; Below: histogram representations corresponding to each level. The final PHOG vector is a weighted concatenation of vectors (histograms) for all levels. Remaining rows: images from the same and from different categories, together with their histogram representations.

a certain angular range. This representation can be compared to the traditional “bag of (visual) words”, where here each visual word is a quantization on edge orientations. A similar representation is used in [15]. We will refer to the representation of each region as a Histogram of Orientated Gradients [36] (HOG).

6.2.2 Spatial layout

In order to introduce spatial information we follow the scheme proposed by Lazebnik *et al.* [76]. In our case, a HOG vector is computed for each grid cell at each pyramid resolution level. So the visual words in chapter 5 are replaced by visual words for a particular orientation. The final PHOG descriptor for the image is a concatenation of all the HOG vectors. In forming the pyramid the grid at level l has 2^l cells along each dimension. Consequently, level 0 is represented by a K-vector corresponding to the K bins of the histogram, level 1 by a $4K$-vector etc, and the PHOG descriptor of the entire image is a vector with dimensionality $K \sum_{l \epsilon L} 4^l$. For example, for levels up to $L = 1$ and $K = 20$ bins it will be a 100-vector. In the implementation we limit the number of levels to $L = 3$ to prevent over fitting.

The PHOG is normalized to sum to unity. This normalization ensures that images with more edges, for example those that are texture rich or are larger, are not weighted more strongly than others.

Figure 6.1 shows that images from the same category have a similar PHOG representation and that this representation is discriminative between categories. Note, PHOG is not the same as a scale space pyramid representation of edges [82] as there is no smoothing between levels of the pyramid, and all edges are computed on the high resolution image.

Similarity between a pair of PHOGs is computed using a distance function, with appropriate weightings for each level of the pyramid. In previous work [57, 76], the distance function was histogram intersection and the weightings were fixed and data independent. In this paper we learn the weightings for the levels, and show that a χ^2 distance has superior performance to histogram intersection.

Figure 6.2: A comparison of difference in PHOG descriptors using χ^2 (gray bars) to Chamfer distance (dark green bars). Note, the y-axis shows $1-$distance, so that a perfect match corresponds to unity, and a poor match to zero. (a) shows a synthetic model image and (b) its matching to the synthetic images (arranged on the x-axis). Similarly, (c) shows a real model image and (d) its matching to images from Caltech-101.

6.2.3 What is being represented?

In order to gain an intuition into what is represented by a difference between PHOG descriptors, we show here its relation to Chamfer distance between the edge maps. Comparing boundaries using Chamfer [19] has proven to be a reliable and efficient method for object recognition e.g. the pedestrian detector of Gavrila & Philomen [52] and the hand tracking of [131]. The chamfer distance between two curves measures the average over the closest distance between them. If the model and target curves are represented by the point sets $\{\mathbf{x}_m\}$ and $\{\mathbf{x}_t\}$ then the Chamfer distance can be computed as:

$$Chamfer = \frac{1}{N_m}\sum_m min_{\mathbf{x}_t}||(\mathbf{x}_m - \mathbf{x}_t)|| \tag{6.1}$$

In addition for curves, rather than point sets, edges are only matched if they have similar orientations, and also distance is capped to reduce sensitivity to “outliers”, such as missed edges through detector drop out [131].

To illustrate the similarity, in figure 6.2 we compare Chamfer distance to the difference between PHOG descriptors computed using χ^2. It can be seen in (b) that both have similar behaviors: both can tolerate missed edges to some extent (second and third examples where an edge is missing) and background clutter (fourth example where an edge is added). PHOG copes better with rotated images due to the additional slack given by computing orientation histograms over regions, whereas Chamfer will cease to find a matching edge that has the same orientation. Figure 6.2(d) shows that for real images (Caltech-101), Chamfer and PHOG again have similar behavior for the most part.

This raises the question of why it is necessary to introduce a new descriptor at all. The answer is that PHOG has three advantages over Chamfer: (i) insensitivity to small rotation (mentioned above). More significantly, (ii) PHOG is a compact vector descriptor suitable for use in standard learning algorithms with kernels. Chamfer matching can be reformulated as a distance between vectors as shown by Felzenswalb [43] (and similarly for Hausdorff matching). However, the vector has the dimension of the number of pixels in the image and will not be particularly sparse; (iii) the principal advantage is that Chamfer matching requires strict spatial correspondence whereas PHOG is flexible, since it builds in spatial pyramid matching, and is able to cope with varying degrees of spatial correspondence by design.

6.3 Modelling shape and appearance

6.3.1 Class specific Level-Weights – CLW

The same P-rbf kernel (section 5.2) is used in an SVM classifier for classifying and retrieving images according to their class (e.g. containing a *motorbike* or a *road*). In previous chapter we globally *learnt* the weights α_l which give the best classification performance over all categories. We propose here to learn the specific level weights for each class. This captures the intuition that some classes are very geometrically constrained (such as a stop sign) whilst others have greater geometric variability (e.g. dolphins, boats). We refer it as CLW – Class specific Level-Weights.

Instead of learning weights common across all classes, the weights α_l are learnt for each class separately by optimizing classification performance for that class using one vs the rest classification. This means that for the 100 categories of Caltech-101 it is necessary to *learn* 400 parameters values (for L=3 levels) instead of only 4 for GLW.

The advantage of learning class-specific level-weights is that classes then have the freedom to adapt if there is more or less intra-class spatial variation, for example. The disadvantage is that the solution is sub-optimal since performance is not optimized over all categories simultaneously. Details and results are given in section 6.6.1.

6.3.2 Merging features

It was shown in section 6.2 that shape represented by PHOG is a good measure of image similarity and thus for image classification. However, shape features alone are not sufficient to distinguish all types of images, as is shown for example by the *strawberry* and *brain* examples in figure 6.2b. In this case, appearance [21, 76] is a better feature to distinguish them. Consequently, we investigate here kernels to combine these two features (shape & appearance). The implementation details are given in section 6.5.

We consider two kernel combinations. The first is a simple linear blend, the second involves a hard choice between the feature types using a $\max$ operation.

The first merging kernel that we propose is based on the weighted sum of appearance and shape information:

$$K(x,y) = \alpha K_A(x_{App}, y_{App}) + \beta K_S(x_{Shp}, y_{Shp}) \tag{6.2}$$

where α and β are the weights for the appearance and shape kernel (5.2) respectively. It has the capacity to give higher weights to the more discriminative features during learning. Moreover it also has the capability to ignore features which do not match well if $\alpha = 0$ or $\beta = 0$. It is a Mercer kernel [61]. Previous authors have considered similar merging kernels, but the optimization differs. We optimize performance over a validation set directly whereas in [6, 80] the interest is in efficient optimization of a proxy for classification performance.

The second merging kernel is based on taking the maximum value of the appearance or shape kernel:

$$K(x,y) = max[K_A(x_{App}, y_{App}), K_S(x_{Shp}, y_{Shp})] \tag{6.3}$$

This kernel has the ability to ignore the appearance or edges features if those features do not match well for a particular exemplar. Note that it is not a Mercer kernel, but this has not proven to be a problem in practice.

For kernel in (6.2) we again consider two learning situations:

- **GFW – Global Feature-Weights**. Optimize weights α and β in (6.2) over all categories together, so that all the categories will have the same weights for the kernel features used.
- **CFW – Class specific Feature-Weights**. Optimize weights α and β in (6.2) for each class separately. Again, these weights are learnt by classifying that class against all others.

6.4 Datasets and methodology

Datasets

We provide results for object classification on Caltech-101 and Caltech-256 and video shot retrieval on TRECVID 2006. See chapter 3 for a complete description.

Figure 6.3 shows examples of what is being represented by the spatial shape descriptor for Caltech-101 dataset. From the average image (averaged over 25 training images) it is evident that images from the same category are very well centered and do not suffer from much rotation. The average gradient and edge images show the strong alignment of the principal object edges within a class – note in particular the alignment of the cup and metronome boundaries. The gradient magnitude weighting of the histogram bins is particularly beneficial here as the strong gradients that are common within a class score highly in the HOG, in turn reinforcing the similarity of PHOGs for images of the same class. It is clear from the averaged orientation histograms for two levels that more local spatial-shape ($l = 3$) is able to distinguish between classes better than global ($l = 0$).

Methodology

Following standard procedures, the Caltech-101 data is split into 30 (25 for training and 5 for the validation set) training images (chosen randomly) per category and 50 for testing – disjoint from the training images. For Caltech-256, 30 images are used for training (25 training and 5 for the validation set) and 25 for testing. For a comparison with [59] for Caltech-256, we report experiments without the last 6 categories and without clutter, this is 250 categories. The final performance score is computed as the mean recognition rate

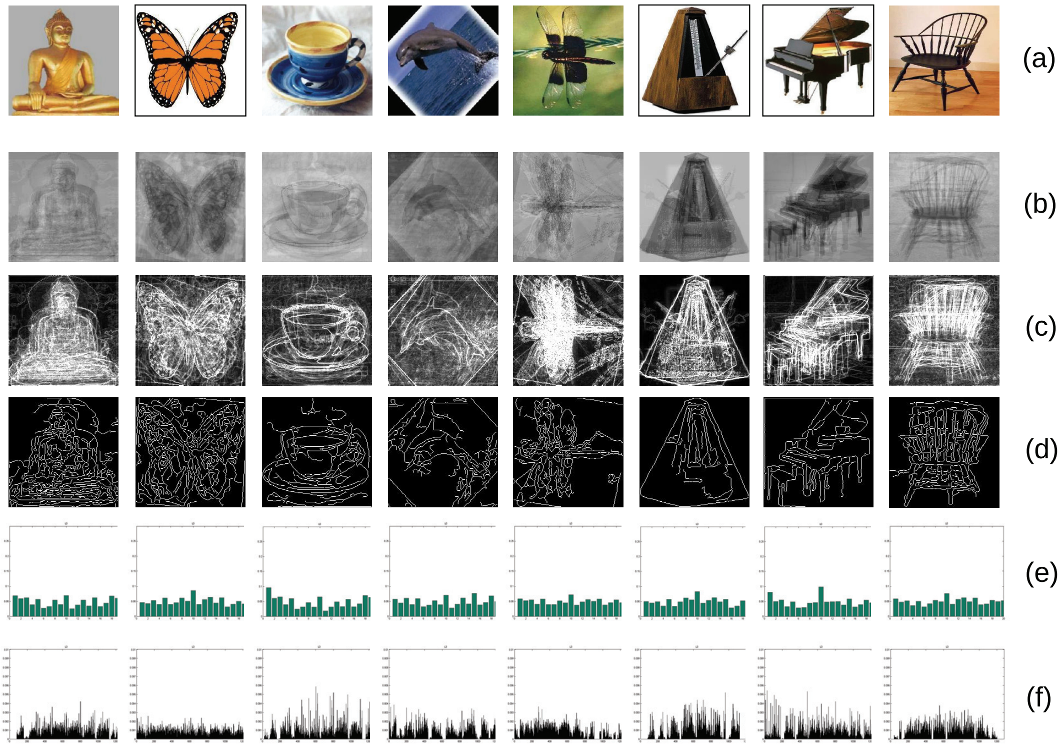


Figure 6.3: Caltech-101 training set: (a) representative image for the class; (b) average over images; (c) average over gradient images; (d) average over edge images; (e-f) orientation histograms at $l = 0$ and at $l = 3$. Note: (i) the gradient and edge images illustrate that a spatial orientation histogram with some tolerance to translation will capture these classes well; (ii) the classes have quite similar global (level $l = 0$) orientation histograms, but differ in their finer spatial layout (level $l = 3$).

per class. The classification process is repeated 10 times, (changing the training and test sets), and the average performance score and its standard deviation are reported.

The validation set is used to optimize all the parameters (e.g. K, α_l). The final performance score is computed as the mean recognition rate per class, so that more populous (and easier) classes are not favored. The classification process is repeated 10 times, (changing the training, validation and test sets), and the average performance score and its standard deviation are reported.

6.5 Implementation

For image classification we use the kernels defined in section 6.3 in a SVM classifier [31]. Multi-class classification is done using one-versus-all SVM: a classifier is learnt to separate each class from the rest, and a test image is assigned the label of the classifier with the highest response. For the retrieval results in TRECVID 2006 we use the probability estimated provided by [31] to rank the representative keyframes (shots).

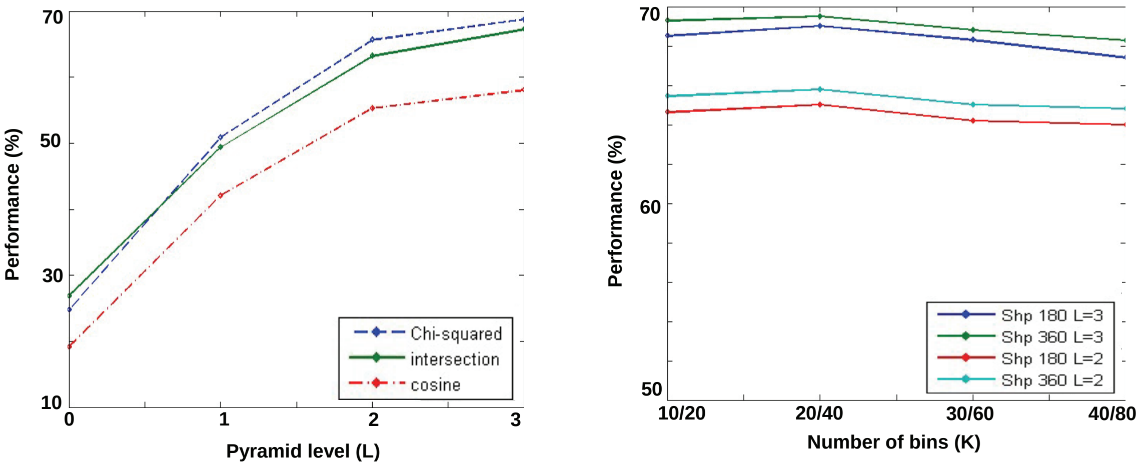
Shape implementation

Edge contours are extracted using the Canny edge detector. The orientation gradients are then computed using a 3×3 Sobel mask without Gaussian smoothing. It has been shown previously [36] that smoothing the image significantly decreases classification performance. The HOG descriptor is discretized into K orientation bins. The vote from each contour point depends on its gradient magnitude, and is distributed across neighboring orientation bins according to the difference between the measured and actual bin orientation. Histograms with K ranging between 10 and 80 bins are tested.

In the experiments two HOG descriptors are compared: one with orientations in the range $[0, 180]$ (where the contrast sign of the gradient is ignored) and the other with range $[0, 360]$ using all orientation as in the original SIFT descriptor [84]. We refer to these as $Shape_{180}$ and $Shape_{360}$ respectively.

Appearance implementation

For the appearance experiments both gray level and colour representations are tested (termed App_{Gray} and App_{Colour} respectively). We follow the approach in chapter 4. SIFT descriptors are computed at points on a regular grid with spacing M pixels, here $M = 10$. At each grid point the descriptors are computed over circular support patches with radii $r = 4$, 8, 12 and 16 pixels. Consequently each point is represented by four SIFT descriptors. Multiple descriptors are computed to allow for scale variation between images. The patches with radii 4 do not overlap and the other radii do. For App_{Colour} the SIFT descriptors are computed for each HSV component. This gives a 128×3 D-SIFT descriptor for each point. In the case of App_{Gray} SIFT descriptors are computed over the gray



(a) (b)

Figure 6.4: (a) Caltech-101 validation set performance for different distance measures over pyramid Levels $L = 0$ to $L = 3$; and (b) over the number of bins (K).

image (with intensity I = 0.3R + 0.59G + 0.11B) and the resulting SIFT descriptor is a 128 vector. Note that the descriptors are rotation invariant. The dense features are vector quantized into *visual words* using K-means clustering. The K-means clustering was performed over 5 training images per category selected at random. A vocabulary of 300 words is used here. Each image is then represented by a histogram of word occurrences (A_I). This forms the feature vector for an SVM classifier, here using the spatial pyramid kernel (5.2).

6.6 Classification results

6.6.1 Parameter optimization

Parameter optimization is carried out on the validation set (disjoint from the training and test set). For example the weights α_l, (5.2) are learnt by maximizing the performance score on the validation set. The optimization is carried out by an exhaustive search over a range of values with granularity 0.1 for α_l, α and β and granularity 10 for K.

Distance measures

We explore here three distance measures: histogram intersection, χ^2, and the normalized scalar product (cosine of angle). For this experiment $\alpha_l = 1$ and $K = 20$ using $Shape_{180}$. The best results are obtained with χ^2 (figure 6.4a), and consequently this distance is used for the rest of this section.

Number of bins – K

We change the value of K in a range $[10\ldots40]$ for $Shape_{180}$ and $[20\ldots80]$ for $Shape_{360}$. Note that the range for $Shape_{360}$ is doubled, so as to preserve the original orientation resolution. Performance is optimal with $K = 20$ orientations bins for $Shape_{180}$, and $K = 40$ for $Shape_{360}$ as shown in figure 6.4b. However, as can be seen, the performance is not very sensitive to the number of bins used.

Level-Weight – α_l

Since the final PHOG is normalized there are only three independent parameters which represent three of the ratios in: α_0:α_1:α_2:α_3. For GLW we optimize the ratios varying them in a range $[0, 2]$. Best performance is achieved with $\alpha_0 : \alpha_3 = 0.2$, $\alpha_1 : \alpha_3 = 0.4$ and $\alpha_2 : \alpha_3 = 0.4$. For CLW we optimize the same ratios and vary their value independently for each category. For the most part more weight is given to $l = 3$, however for *accordion* and *inline-skate* (and some others) more importance is given to $l = 2$ or $l = 1$. This is because at the higher levels regions are smaller and may miss the object entirely (capturing background) as the object position varies in the image (there is more intra-class spatial variation). An example is shown in figure 6.5 with the accordion category.

Kernel features – α & β

For the GFW we learn the kernel weights in (6.2) by varying the ratio $\alpha : \beta$ in the range $[0\ldots2]$. In this case best performance is obtained with α:$\beta = 1.5$. This means that appearance has more influence. For CFW optimization there exist categories for which appearance works better (e.g. *leopards, barrel, sail boat*) and others for which shape is

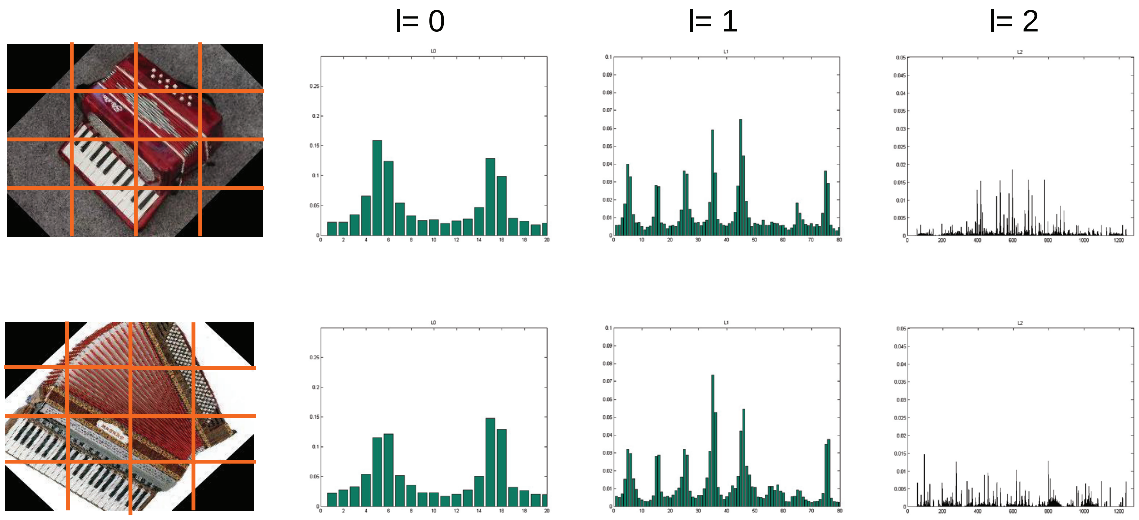


Figure 6.5: Object level overfitting for the accordion category, in this case more weight is given to lower pyramid levels ($l = 0$ and $l = 1$) than to the higher ones ($l = 2$ or $l = 3$)

best (e.g. *scissors*, *lamp*, *yin-yang*, *umbrella*). Figure 6.6 shows some examples.

6.6.2 Caltech-101

Shape alone

The first four rows in table 6.1 summarize the performances achieved using just the HOGs at one level of the pyramid (single level) as well as the performances when using the PHOG for GLW optimization. As can be seen in this table (and in figure 6.4a), very poor performance is obtained for the lowest level ($L = 0$). In this case we are representing the images with just one orientation histogram and this is not very discriminative between classes (see figure 6.3e). However performance increases by around 40% when introducing spatial information, so the PHOG becomes far more discriminative (see figure 6.3f). This means that objects can be more easily differentiated by their spatially local orientations than by global measures. Though matching at the highest pyramid level seems to account for most of the improvement, using all the levels together confers a statistically significant benefit.

Using $Shape_{360}$ we obtain the best result (69.0%). Including contrast sign information helps substantially for recognition tasks in Caltech-101, because this sign information is

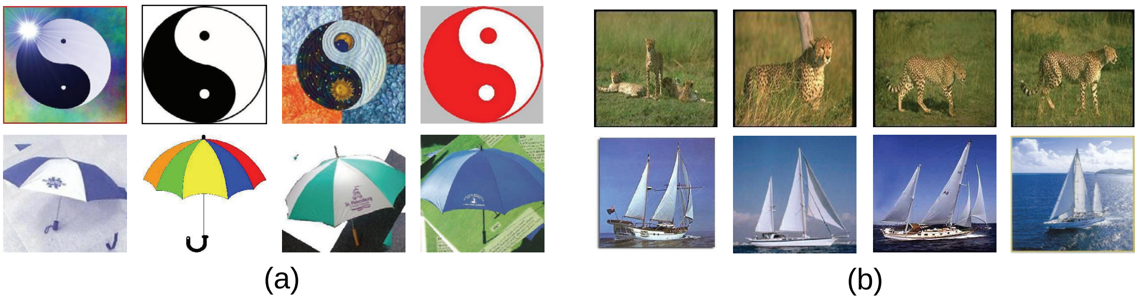


Figure 6.6: Intra-colour variability for a few different categories on Caltech-101. (a) some images for which shape is better; (b) some images for which colour works better.

	Single level			
	l=0	l=1	l=2	l=3
S_{180}	23.2 ± 0.5	47.3 ± 0.7	61.7 ± 0.5	64.3 ± 0.8
S_{360}	25.0 ± 0.4	49.4 ± 0.7	62.1 ± 0.7	66.9 ± 0.9
	Pyramid – PHOG			
	L=0	L=1	L=2	L=3
S_{180}	23.2 ± 0.4	49.3 ± 0.6	64.1 ± 0.6	67.2 ± 0.5
S_{360}	25.0 ± 0.5	51.4 ± 0.8	64.2 ± 0.7	$\mathbf{69.0} \pm 0.6$
A_G	55.3 ± 0.4	64.6 ± 0.3	67.0 ± 0.5	$\mathbf{68.1} \pm 0.6$
A_C	52.2 ± 0.5	63.1 ± 0.7	65.3 ± 0.9	66.5 ± 0.8

Table 6.1: Caltech-101 performance using appearance and shape separately with a χ^2 kernel. Single level means that only a HOG from level l is used. For PHOG, GLW is used to find the α_l. Note that S_{360} has slightly better performance than A_G.

useful for the majority of objects in the dataset, like *motorbikes* or *cars*. This is in contrast to [36] where better results were obtained if the contrast sign of the gradient was ignored. This is because [36] is detecting pedestrians, and in this case the wide range of clothing and background colors presumably makes the contrast sign uninformative. If we use the CLW (class specific level-weight optimization) the score increases as far as 69.8% for $Shape_{180}$ and, to 70.6% for $Shape_{360}$.

Figure 6.7 samples some examples of class confusions, and it is evident that the confusion is understandable, and arises here (and in others we have looked at, but have not included) from shape similarity, for example the *soccer ball* and *watch* in figure 6.7a are

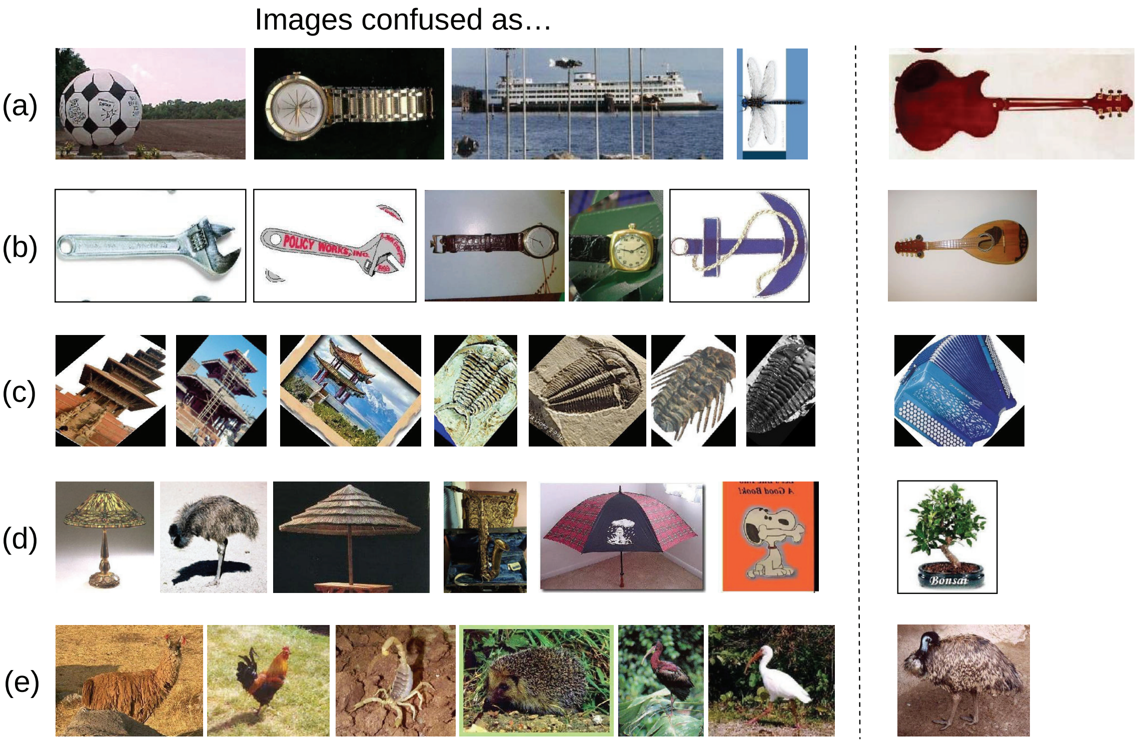


Figure 6.7: Examples of categories that are confused using shape alone: (a) images confused with a guitar; (b) images confused with a mandolin; (c) images confused with an accordion; (c) images confused with a bonsai; (d) images confused with an emu.

both confused with a *guitar*. This is because all three images have a circular part to the left of the image, and two straight lines to the right. Figure 6.7c shows the *pagoda* and *trilobite* being confused with an *accordion* (due to the rotation and the small lines in the image), and in figure 6.7d the *lamp* and the *emu* are confused with a *bonsai*.

Figure 6.8 samples the contour image and PHOG descriptor of some confused images with accordion category. Note that all the contour images have lots of edges with the same direction and consequently they have very similar edge orientation histograms. Figure 6.9 samples some images confused with bonsai category. Again all the contour images have very similar edges (with the same orientations) and the PHOG for all these images is also very similar. Images in figure 6.10 are some images confused with emu category. Note that in the contour images the shape of the objects can not be appreciated and in this case the background edges are causing the confusions. All these images have vegetation at the background and this is what is captured by the PHOG descriptor.

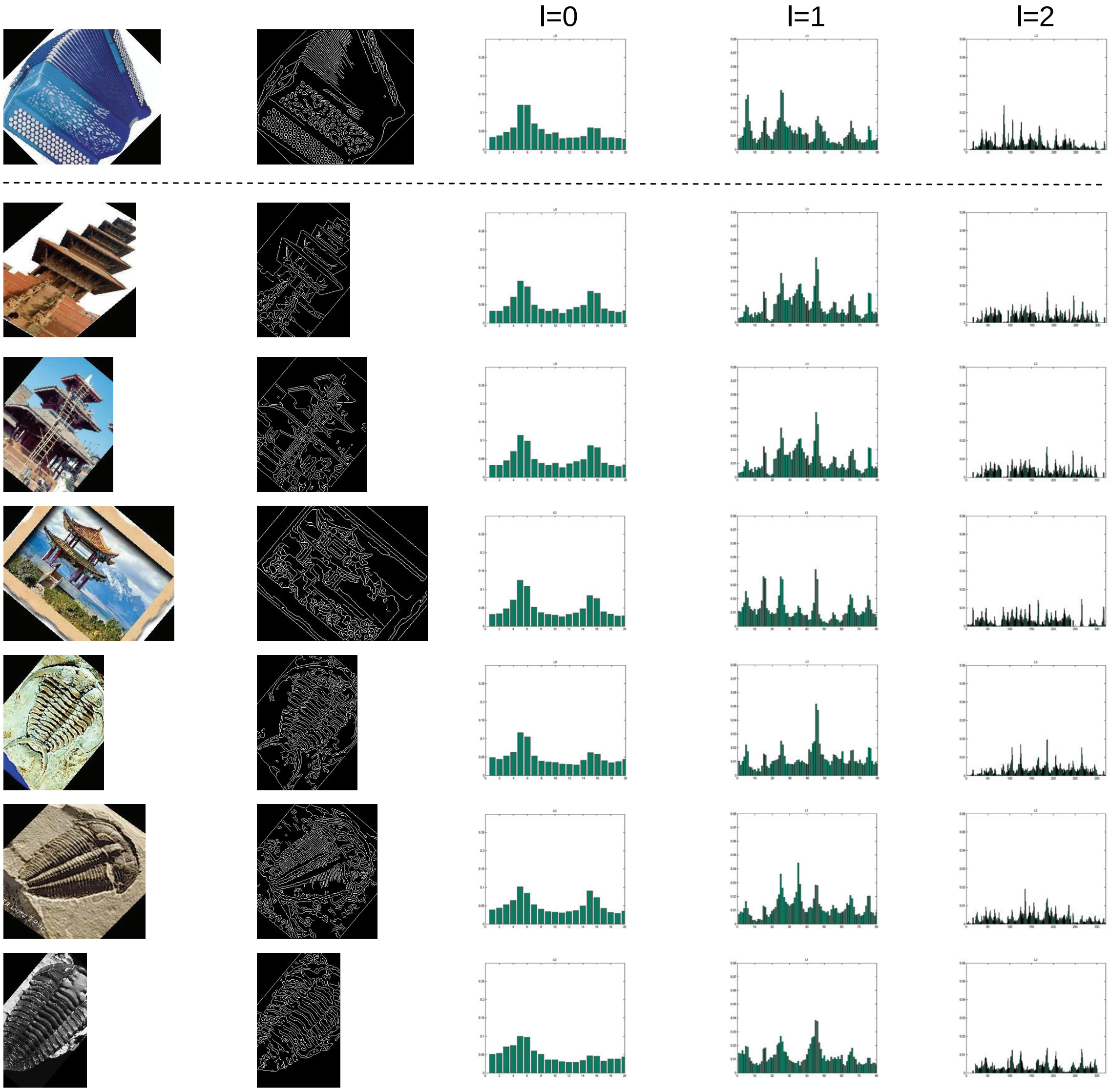


Figure 6.8: PHOG of some confused images with accordion.

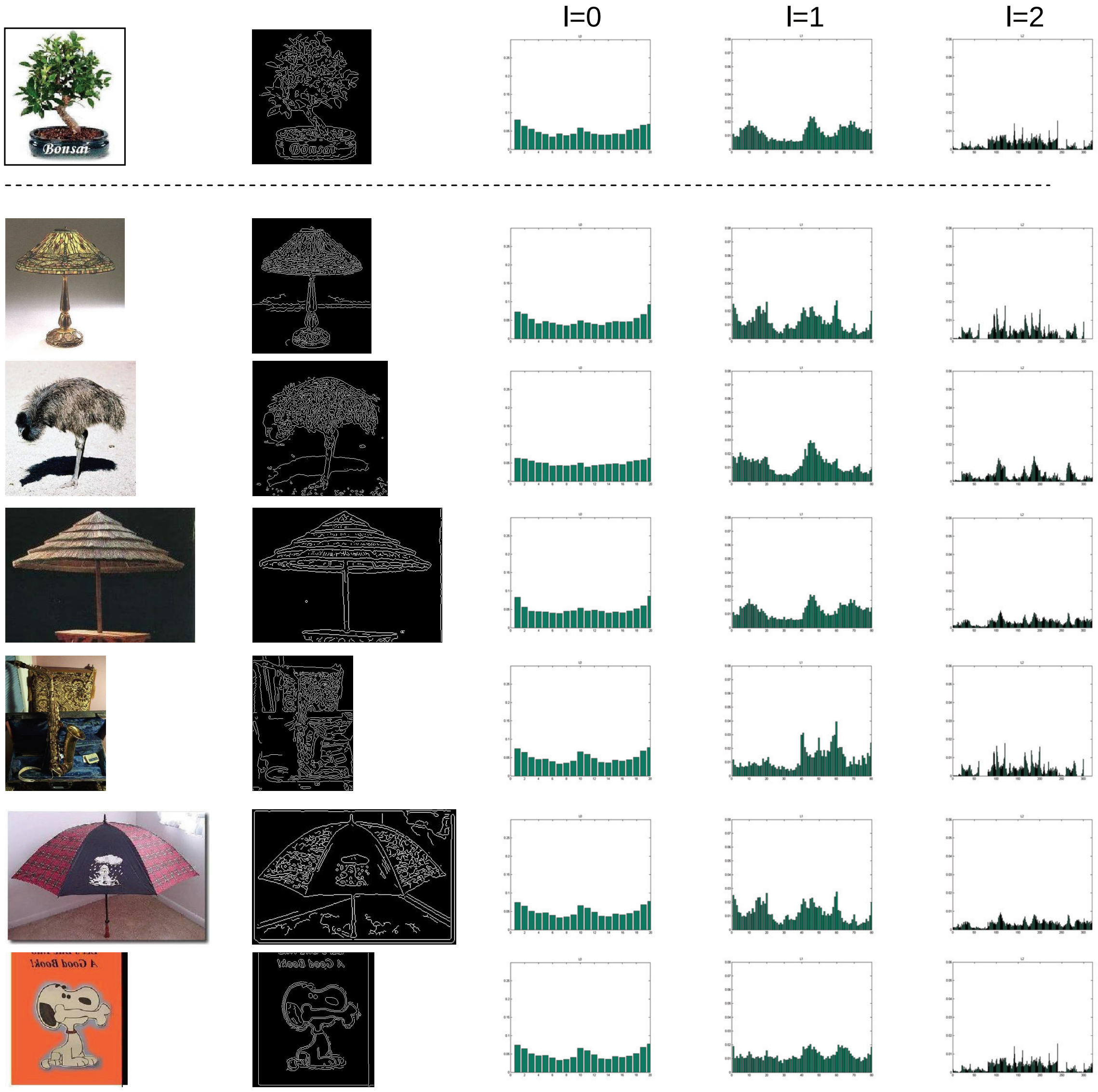


Figure 6.9: PHOG of some confused images with bonsai.

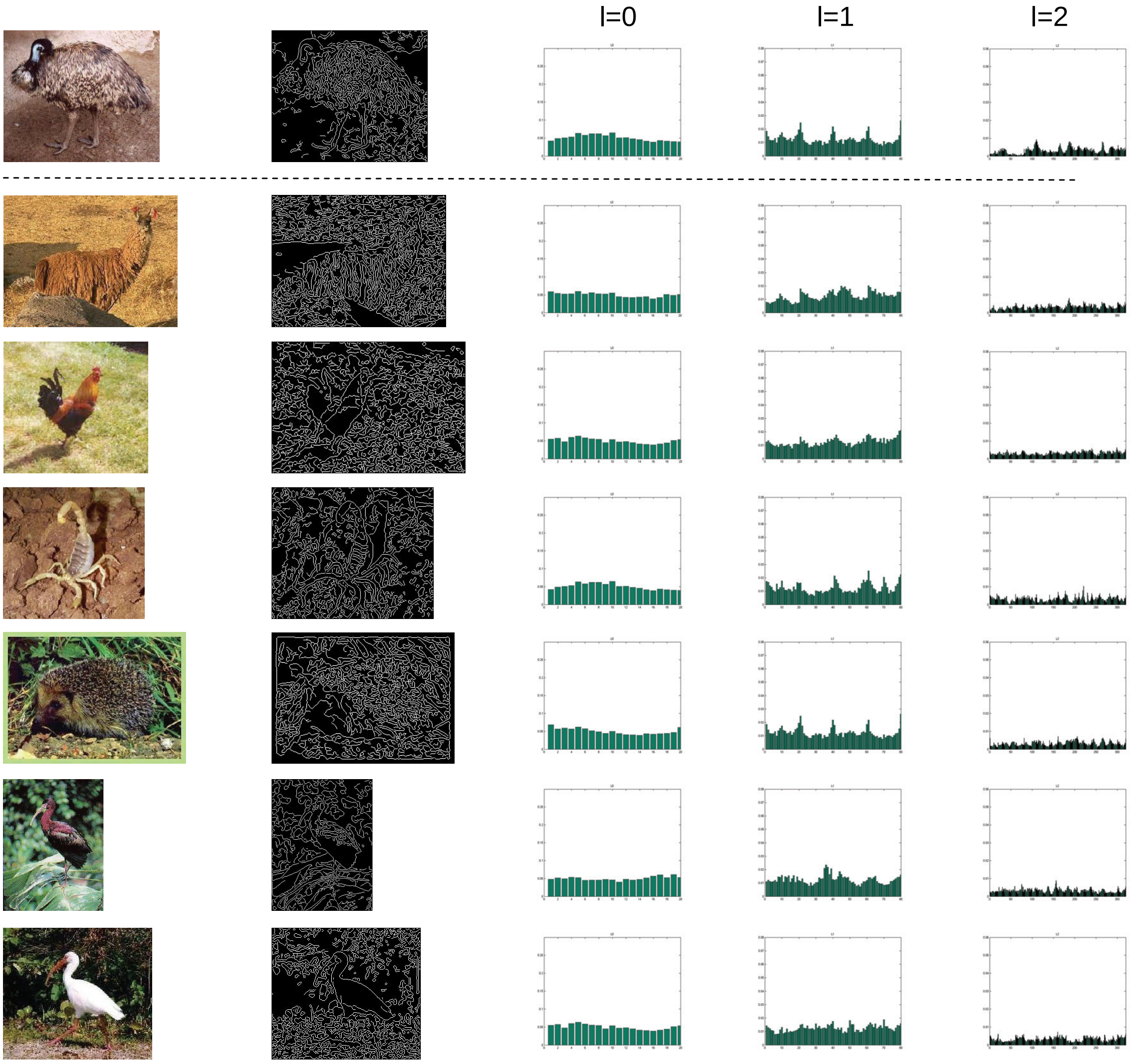


Figure 6.10: PHOG of some confused images with emu.

The best classified classes are *octopus*, *metronome* and *inline skate* with a score of 100% and the worst are *brontosaurus* and *cannon*, with 25.0% and 25.7% respectively.

Appearance alone

The last two rows of table 6.1 summarize the appearance performance for GLW optimization. For Caltech-101 App_{Gray} works better than App_{Colour} (68.1% vs 66.5%). If we use CLW then the score increases to 71.6% for App_{Gray} and to 68.2% for App_{Colour}. However, it is not a true rule for all the categories. In some individual categories colour information is very relevant – and are less confused when using it – (e.g. ketch, leopards) whilst for others it is not (e.g yin-yang, umbrella). Figure 6.6 shows examples of each case.

Shape & Appearance

We first use the kernel in (6.2) with GLW and GFW. When merging App_{Gray} and $Shape_{180}$ the performance is 70.7%, and this increases to 71.5% when merging App_{Gray} and $Shape_{360}$. For GLW and CFW performances increase to 72.8% for App_{Gray} and $Shape_{180}$, and to 73.5% for App_{Gray} and $Shape_{360}$. The best results are obtained using both class-specific optimizations (CLW & CFW): 76.2% for App_{Gray} and $Shape_{180}$, and 76.6% for App_{Gray} and $Shape_{360}$. That merging with $Shape_{360}$ is better than with $Shape_{180}$ is expected, since $Shape_{360}$ alone performs better. Using the kernel in (6.3) and class-specific optimization, performances are slightly improved at 76.6% and 76.7%.

We have at our disposal two kind of appearances cues (App_{Gray} and App_{Colour}) and two kinds of shape representation ($Shape_{180}$ and $Shape_{360}$). If we merge all the kernels representing these cues using CLW and CFW, then we obtain the best performance overall: 77.8%. Table 6.2 gives a summary of the performances using different feature combinations. For just one feature CLW is used. Both class-specific optimizations (CLW & CFW) are used for merging cues. If we use the kernel defined in (6.3) to merge all the appearance and shape features, then a slightly worse result of 77.5% is obtained. The conclusion is that the kernel in (6.2) works better than the kernel in (6.3) when a large number of kernel features are used.

$Shape_{180}$	$Shape_{360}$	App_{Gray}	App_{Colour}	Perform
✓				69.8 ± 0.5
	✓			70.6 ± 0.6
		✓		71.6 ± 0.6
			✓	68.2 ± 0.8
		✓	✓	75.3 ± 0.6
✓		✓		76.2 ± 0.6
	✓	✓		76.6 ± 0.8
✓	✓	✓	✓	77.8 ± 0.8

Table 6.2: Comparison rates when using different cues on Caltech-101 dataset: ✓ means the cue is used. Feature selection is carried out using the optimization process CLW and CFW. The kernel defined in (6.2) is used for merging cues.

[100]	[58]	[151]	[76]	[157]	Ours
56.0	58.23	63.0	64.6	66.0	77.8 ± 0.8

Table 6.3: Classification of Caltech-101 with 30 training images per class.

Comparison to previous results

Table 6.3 compares our results to those reported by other authors. Our best performance is 77.8%. This outperforms the results reported by Zhang *et al.* [157] that to our knowledge were the best until now. The merging appearance and shape features together with the optimizations process improves performance by 10% compared to the results obtained in previous chapter.

6.6.3 Caltech-256

Performances for this dataset with the two class-specific optimizations (CLW & CFW) are summarized in table 6.4. These results are for the first 250 categories in this dataset and our best result is when merging all the appearance and shape features obtaining an score of **40.1%** of correctly classified images. For the whole dataset, without clutter category, our performance is 39.6% using both class-specific optimizations and all the appearance and shape cues. Note that performance on Caltech-256 is roughly half the performance

$Shape_{180}$	$Shape_{360}$	App_{Gray}	App_{Colour}	Perform
✓				34.9 ± 0.6
	✓			37.1 ± 0.8
		✓		37.5 ± 0.8
			✓	32.3 ± 0.7
✓	✓	✓	✓	40.1 ± 0.8

Table 6.4: Comparison rates when using different cues on Caltech-256 dataset: ✓ means the cue is used. Feature selection is carried out using the optimization process CLW and CFW. The kernel defined in (6.2) is used for merging cues.

achieved on Caltech-101: for 30 training images our performance on Caltech-101 and Caltech-256 are 77.8 ± 0.8 and 40.1 ± 0.8 respectively.

Some of the most confused categories in Caltech-256 are those which are closely related. For example *tennis shoes* and *sneakers*, *frog* and *toad*. Figure 6.11 shows some examples. Images on the left show confused images with the category on the right, and images on the right are confused images with the category on the left. The percentage is the miss-classification rate of Class A as Class B and viceversa. For example in the first row, the three first *sneakers* images are confused as *tennis shoes* with a miss-classification rate of 32%, and the three *tennis shoes* images are confused as *sneakers* with a miss-classification rate of 35%. Note that all of them are related classes with very similar shape and appearance.

Comparison to previous results

Griffin *et al* [59] have a performance of 34.1% using the PHOW descriptor with the method in [76]. We increase the performance up to 40.1% that to our knowledge is the best until now for this dataset.

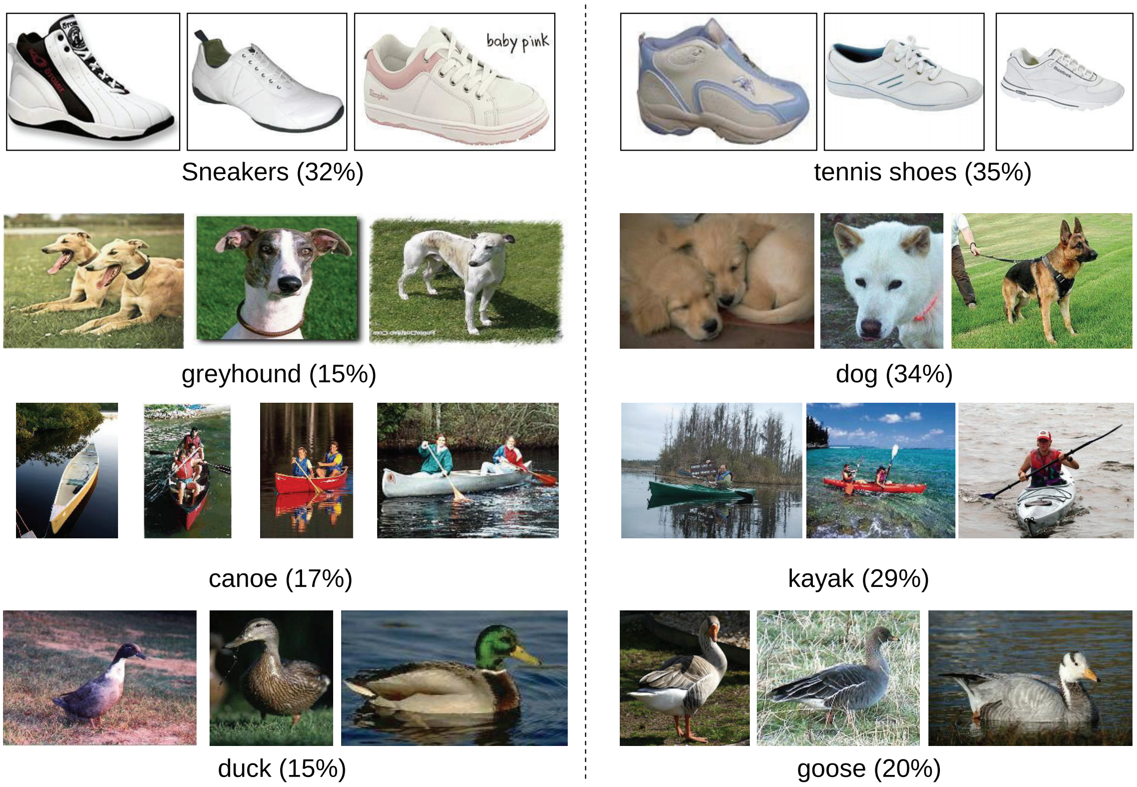


Figure 6.11: Related confused images on Caltech-256 using both appearance and shape descriptors and both class-specific optimizations CLW and CFW. Images on the left are images confused as class on the right and images on the right are images confused as class on the left. The percentage is the miss-classification rate.

6.7 Retrieval results – TRECVID

Methodology on the training data

For the TRECVID data we used the most representative keyframe of each video shot (43907 keyframes). This data is also split into three disjoint sets: 27093 keyframes for training, 3900 (100 per category) keyframes for the validation set and 12914 keyframes for testing. In this case precision vs. recall graphs are computed and the average precision is reported as a performance measure, following the TRECVID guidelines.

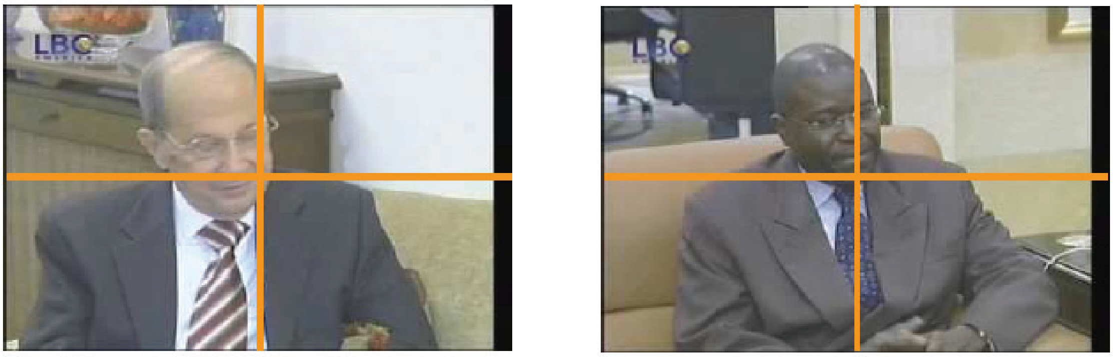


Figure 6.12: Spatial information used for the scene representation on TRECVid 2006 dataset. Each dimension of the image is divided by two obtaining 4 bins.

Methodology on the test data

For each class, we used all the positive training shots and 10000 randomly selected shots as negative examples from the training data supplied by MediaMill. However some categories were given special treatment:

- Snow: We also used shots which have a large amount of white colour as negative examples.
- Airplanes: We also used golf shots as negative examples. Some scenes with small airplanes in the sky are very similar and often confused with golf scenes with a small ball in the middle of the grass.

Implementation

The appearance representation is computed in a similar manner to that described in section 6.5. A vocabulary of 1500 words is used for App_{Colour}. The number of bins for PHOG is set to $K = 40$ using $Shape_{360}$ (we have not optimized K here, as it is demonstrated in section 6.6.1 that it does not have much affect on the final results). Level weights (α_l) and kernel weights (α and β) are learnt on the validation set. We used χ^2 distance together with a spatial pyramid kernel with $L = 1$, which means that the inputs for the discriminative classifiers have a dimension of $1500 \times 5 + 20 \times 5$. In TRECVID spatial

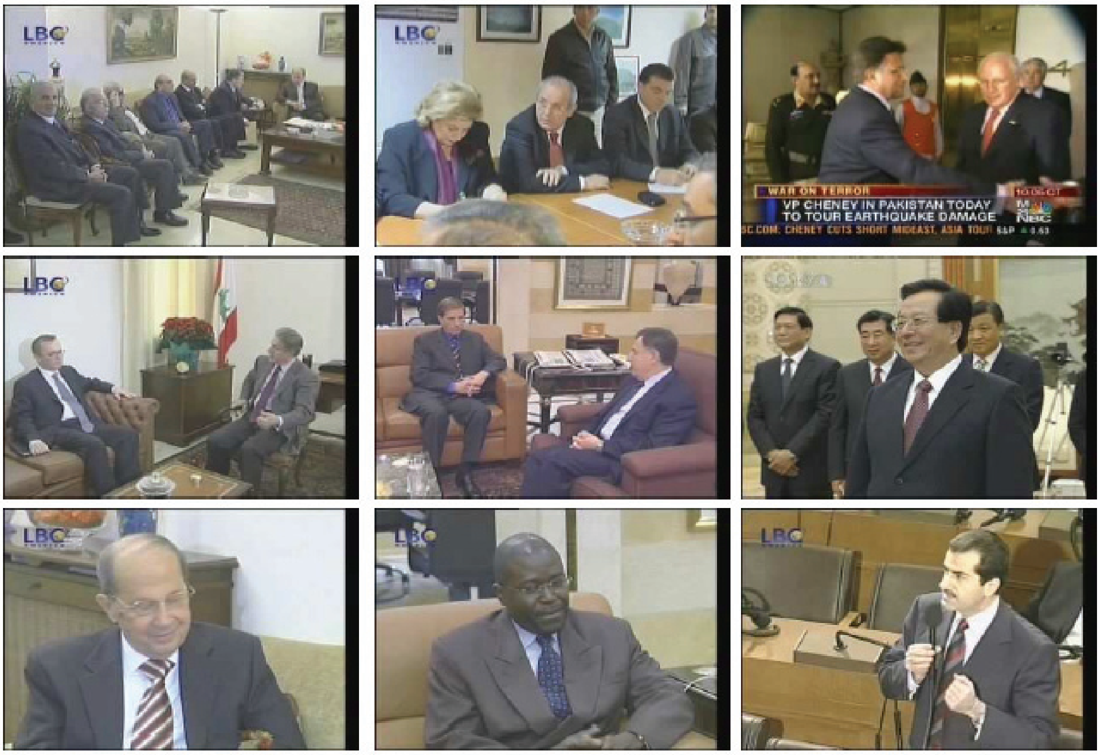

WAR ON TERROR
VP CHENEY IN PAKISTAN TODAY
TO TOUR EARTHQUAKE DAMAGE
SC.COM: CHENEY CUTS SHORT MIDEAST, ASIA TOUR

Figure 6.13: Examples of scene matching for the topic "corporate leaders". Spatial layout is important for such cases as people are often framed in a common manner/pose for single individuals, pairs, small groups etc.

distribution is very variable between images of the same category. Consequently we only used up to $L = 1$ for this data set. The spatial layout is illustrated in figure 6.12.

Results on training data

We learn the appearance and shape classifiers for all 39 required topics. In general, most of the categories are best retrieved when using App_{Color} features alone (e.g *face* – AvPr = 99.4%, *Weather* – AvPr = 95.3%, *Sky* – AvPr = 91.6%). However, there are some categories like *truck* or *screen* where $Shape_{360}$ works better. Concretely there is a significant increase on performance over App_{Colour} for *building* (from 7.8% for App_{Colour} to 44.5% for $Shape_{360}$) and *road* (from 5.4% to 18.6%). Best results are obtained when using the appearance and shape merging kernel (6.2) with both class-specific optimizations (CLW & CFW). In this case there is a significant performance improvement when retrieving:

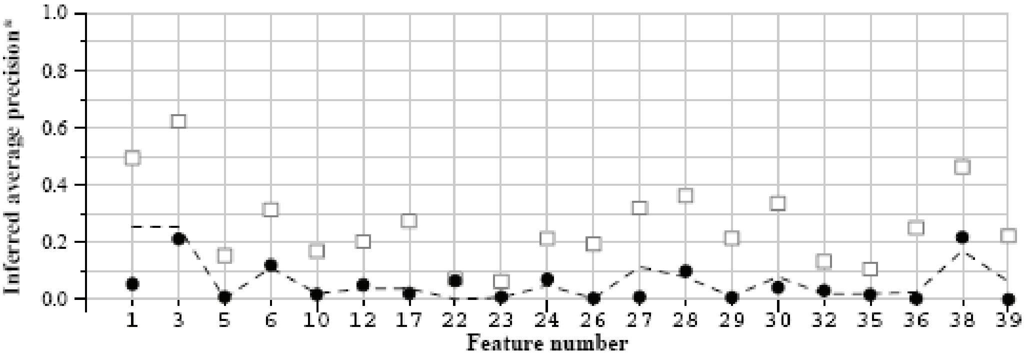


Figure 6.14: This graph represents the average precision of our retrieved shots run score (dot) versus median (–) versus best (box) by topic. See chapter 3 for a correspondence topic–feature number.

crowd – 84.2%, *entertainment* – 74.9%, *people walking* – 65.4%, *sports* – 61.1%, *mountain* – 59.3% and *military* – 49.9%.

Results on test data

Test data is not annotated and we can not provide quantitative results. We participated on the TRECVid workshop 2006 [114] and we had the first position when retrieving the *corporate leader* category (using only appearance information). Figure 6.13 shows 9 matched shots for the *corporate leader* category, retrieved from the entire TRECVID 2006 test collection when using only appearance information. These scenes have a similar layout. For example: a person centred in the middle of the image, or a two person meeting with a background wall containing some pictures. In all these cases if the image is divided into four bins (as shown in figure 6.12), the similarity of the information in each of them is evident. This shows, somewhat surprisingly, that spatial information is an important feature for such categories.

Figure 6.14 shows the average precision of our retrieved shots (dot) versus median (—) versus best (box) by topic. Note that not all the topics were evaluated for the TRECVid. We can see that our result for topic 22 (*corporate leader*) matches with the best result and in general we are over the median. Our results are above the median for topics 24 (*military*), 28 (*flag-US*) and 38 (*maps*) and under the median for topics 1 (*sports*), 27 (*TV-screen*). *Sports* category include all kind of sports (*basket*, *tennis*, *football*) and they

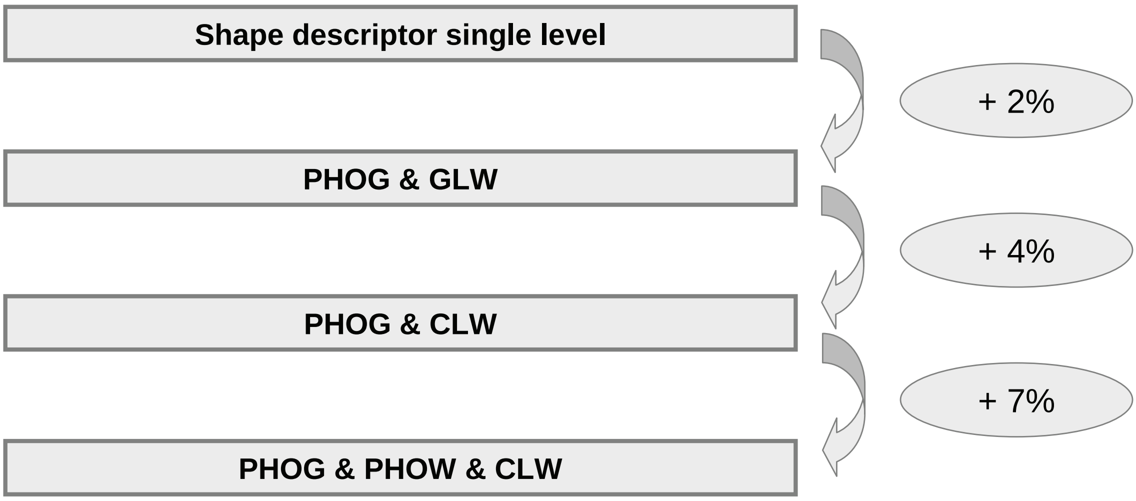


Figure 6.15: This graphs shows how much each representational and feature increment contributes to the overall performance.

do not follow the same spatial layout. The same happens with the *tv-screen* category: screens are not situated in the same position over the images, thus the spatial layout is neither useful for this topic.

6.8 Summary

We have introduced a new descriptor, PHOG, which flexibly represents the spatial layout of local image shape. We can attribute how much each representational and feature increment contributes to the overall performance. Compared to using a single level ($l = 3$) shape representation, PHOG increases performance by 2% using weightings common across all classes. This increase rises to 4% using class-specific level weighting. The combination of the PHOG and appearance descriptors achieves an 11% improvement (compared to the single level shape representation) using the class-specific feature kernel. This demonstrates that the shape and appearance descriptors are complementary. Figure 6.15 shows graphically how much each improvement contributes to the final performance. We conclude that complementary spatial pyramid based descriptors, together with class-specific optimization of pyramid weights and class-specific kernel selection for merging are all important for good performance.

We outperform the state of the art for Caltech-101 and Caltech-256 datasets by 11% and 6% respectively.

6.9 Discussion

Obtained results are very encouraging, however there is still some things to take into account. If we have a look to images from Caltech-256 we can see that there is some translation, scale-variance and background-variance amongst objects from the same category. If we apply the pyramid kernel to the whole image we fail for three reasons: (i) we are not capturing only the object information but the object information and the background as well, so if two objects have a similar background they can be confused; (ii) the classifier relies not just on the presence of individual parts but also on their relationship, if the objects are translated the classifier is not capturing the same object parts and consequently is not capturing the same relationship; and (iii) the classifier has limited scale-invariance: objects or pieces of objects are no longer recognized if their size changes by an order of magnitude, because the pyramid representation is not the same.

Figure 6.16 shows some examples of the three mentioned problems. Figure 6.16a shows three images from different categories whit a similar background. The pyramid descriptor captures the background information and the final histogram is very similar for both images causing a confusion between them. Figure 6.16b shows two images with the same translated object category. Due to this translation the same parts of the object fall in different pyramid bins, and this causes a different descriptor for the same object class. Figure 6.16c shows two images with the same object in different scales. Again the descriptor is capturing different information for the same pyramid bins causing a different descriptor for the same object class.

A way to solve this is using a sliding window classifier. These methods involve training a classifier, which for a small image patch, decides whether the desired object is present. Given a test image, such a classifier is then applied within a “sliding window”, over a range of translations and scales. Extracted image features (image measurements), and the form of the classifier, vary considerably. Training the classifier usually requires many tightly cropped training images (with both object present and absent). The task of the classifier is to capture the intra-class variations present in the training data. Some

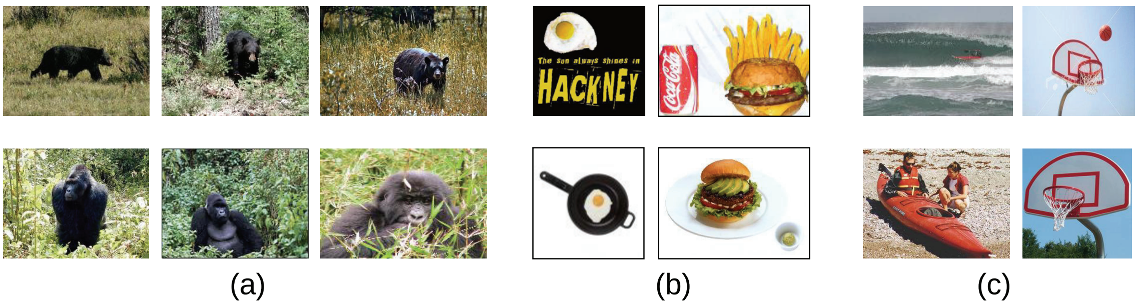


Figure 6.16: This figure shows some problems about using the pyramid descriptor to the whole image. (a) objects with similar background can be confused; (b) objects at different location have different descriptors; (c) objects at different scales have different descriptors.

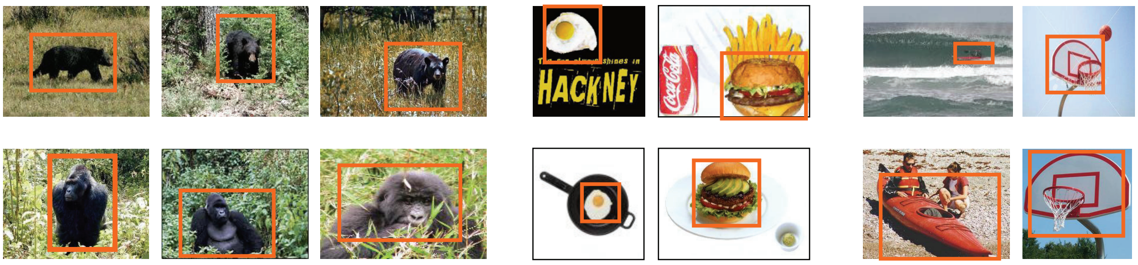


Figure 6.17: The red patch represents the part of the image which contains the objects. Using only this patch to compute the pyramid descriptor will provide a better representation for the objects without taking the background into account. In this case translations and scale variances can be avoided.

examples of sliding window approaches can be found in [93, 122, 139, 148]. Then the descriptor will be computed only for the patch representing the objects and the problems above mentioned will be *solved* because the same objects parts will be captured for the same bins. This is illustrated in figure 6.17 and explained in detail in the following chapter.

Another aspect to tackle is the expensiveness of the method. Now, to recognize-256 objects our system takes around 60 hours to run the M-SVM classifier. If we increase the number of object categories to classify the computational cost is also increasing. An ideal computer vision system should be able to recognize the same number of objects that exist in the real world and it is believed that humans can recognize between 5000 and 30000 object categories [14, 40]. This is without the feature extraction and weight optimization which can be done off-line. So that we also need to think about faster techniques without losing accuracy [90].

Chapter 7

Image classification using random forests and ROIs

In this chapter, we explore the problem of representing the object categories when high translation is presented looking for a trade-off between accuracy and computational time. To this end we combine four ingredients: (i) shape and appearance representations that support spatial pyramid matching over a region of interest. This is a representation common across all classes. It generalizes the PHOW and PHOG representations introduced in chapter 5 and chapter 6 from an image to a region of interest (ROI); (ii) automatic selection of the regions of interest in training. This provides a method of inhibiting background clutter and adding invariance to the object instance's position; (iii) we use some extra training information generated synthetically by rotating, translating and scaling the ROI in the original training data. In this way a more robust training set is obtained; and (iv) the use of random forests (and random ferns) as a multi-way classifier. The advantage of such classifiers (over multi-way SVM for example) is the ease of training and testing.

Results are reported for classification of the Caltech-101 *and Caltech-*256 *data sets. It is shown that selecting the ROI adds about* 5% *to the performance. Together with the extra training data generation the result is about a* 13% *improvement over the state of the art for Caltech-*101 *and a* 10% *improvement over the state of the art for Caltech-*256*. We also compare performance with a benchmark multi-way SVM classifier.*

7.1 Introduction

The release of challenging data sets with ever increasing numbers of object categories, such as the recent Caltech-256 [59], is forcing the development of image representations that can cope with multiple classes and of algorithms that are efficient in training and testing. In this chapter we build on both ideas: the image representation and an efficient classifier.

First the image representation: we improve on the PHOW (see chapter 5) and PHOG (see chapter 6) representations in two ways. For training sets that are not as constrained in pose as Caltech-101 or that have significant background clutter, treating image classification as scene matching is not sufficient. Instead it is necessary to "home in" on the object instance in order to learn its visual description [33]. To this end we automatically learn a Region Of Interest (ROI) in each of the training images. The idea is that between a subset of the training images for a particular class there will be regions with high visual similarity (the object instances). These regions can be identified from the clutter by measuring similarity using the spatial pyramid representation, but here defined over a ROI rather than over the entire image. The result is that "clean" visual exemplars [12] are obtained from the pose varying and cluttered training images. We represent both the appearance (using dense vector quantized SIFT descriptors) and also local shape (using a distribution over edges). These features are applied to a spatial pyramid over a ROI, rather than over the image as in previous chapters. This idea is illustrated in figure 7.1.

Turning to the classifier, we employ here a random forest classifier. These classifiers were first introduced in [4] and developed further in [23]. Their recent popularity is largely due to the tracking application of [78]. They have been applied to object recognition in [97, 153] but only for a relatively small number of classes. Here we increase the number of object categories by an order of magnitude (from 10 to 256). The research question is how to choose the node tests so that they are suited to spatial pyramid representations and matching. The advantage of randomized trees, as has been noted by previous authors [154], is that they are much faster in training and testing than traditional classifiers (such as an SVM). They also enable different cues (such as appearance and shape) to be "effortlessly combined" [153]. The novelty is that at each test node the classifier has the ability to choose the weight given to shape, appearance and the levels of the pyramids without an optimization process over the validation set. This facilitates repre-

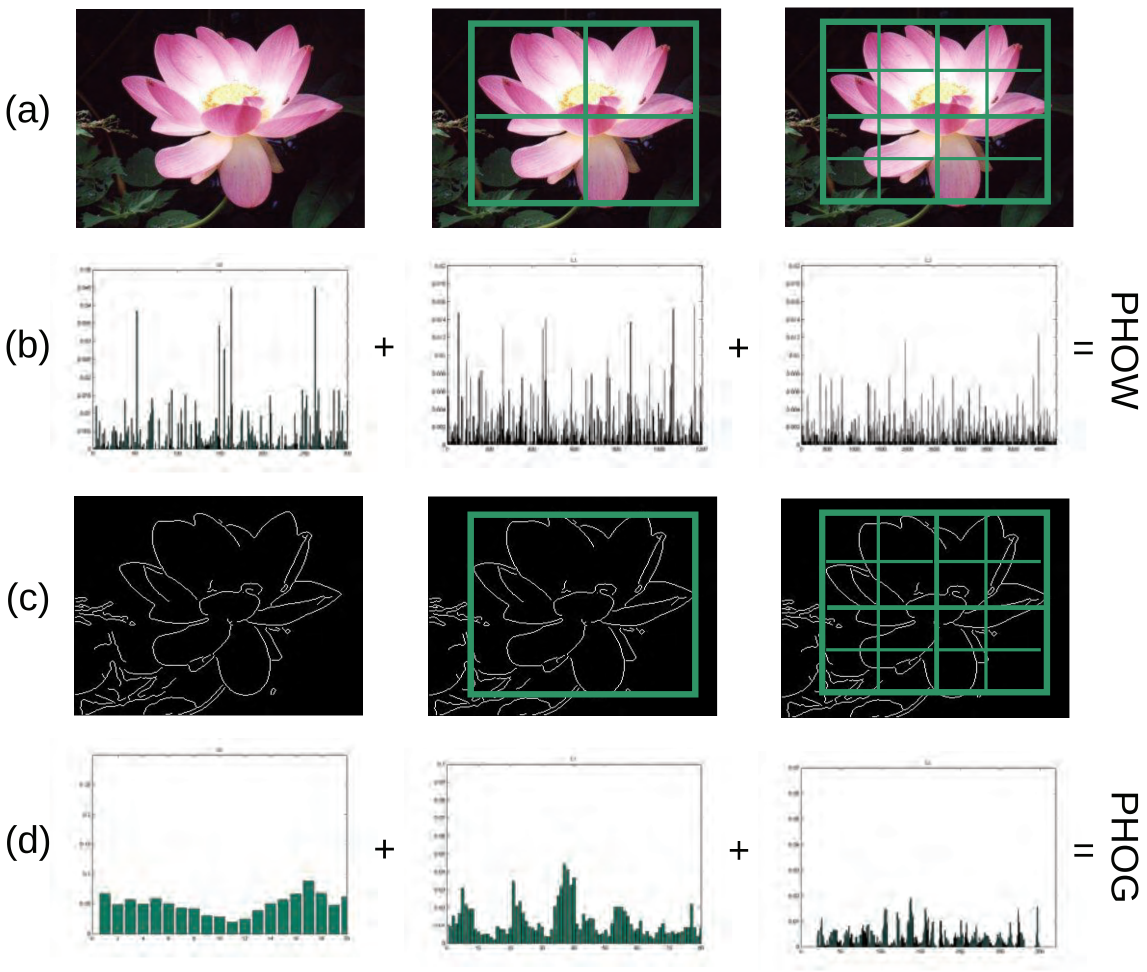


Figure 7.1: Appearance and shape spatial representation. (a,c) Grids for levels $l = 0$ to $l = 2$ for appearance and shape representation over the automatic detected ROI; (b,d) appearance and shape histogram representations corresponding to each ROI level.

sentations and classification suited to the class and is much faster than the weight learning introduced in previous chapter.

In section 7.2 we describe how the ROIs are automatically learnt, and in section 7.3 and section 7.4 how they are used together with random forests (and ferns) to train a classifier. A description of datasets and the experimental evaluation procedure is given in section 7.6. Implementation details are given in 7.7. section 7.8 reports the performance on Caltech-101 and Caltech 256, as well as a comparison with the state of the art. The paper concludes with a discussion and conclusions in section 7.10.

7.2 Selecting the Regions of Interest – ROI

Caltech-256 (and several other datasets used for object recognition, such as PASCAL) has a significant variation in the position of the object instances within images of the same category, and also different background clutter between images (see figure 7.2). Instead of using the entire image to learn the model, an alternative is to focus on the object instance in order to learn its visual description as it is shown in figure. To this end we describe here a method of automatically learning a rectangular ROI in each of the training images. The intuition is that between a subset of the training images for a particular class there will be regions with high visual similarity (the object instances). It is a subset due to the variability in the training images – one instance may only be similar to a few others, not to all the other training images. These "corresponding" regions can be identified from the clutter by measuring their similarity using the PHOW and PHOG described in chapters 5 and 6 but here defined over a ROI rather than over the entire image.

Suppose we know the ROI r_i in image i and the subset of s other images j that have "corresponding" object instances amongst the set of training images for that class. Then we could determine the corresponding ROIs r_j of images j by optimizing the following cost function:

$$\mathcal{L}_i = \max_{\{r_j\}} \sum_{j=1}^{s} K(D(r_i), D(r_j)) \tag{7.1}$$

where $D(r_i)$ and $D(r_j)$ are the descriptors for the ROIs r_i and r_j respectively, and their similarity is measured using the kernel defined by (5.2). Here we use a descriptor formed by concatenating the PHOG and PHOW vectors. As we do not know r_i or the subset of other images we also need to search over these, i.e. over all rectangles r_i and all subsets of size s (not containing i). This is too expensive to optimize exhaustively, so we find a sub-optimal solution by alternation: for each image i, fix r_j for all other images and search over all subsets of size s and in image i search over all regions r_i. Then cycle through each image i in turn. The value for the parameter s depends on the intra-class variation and we explore its affect on performance in section 7.8.

In practice this sub-optimal scheme produces useful ROIs and leads to an improvement in classification performance when the model is learnt from the ROI in each training image. Figure 7.2 shows examples of the learnt ROIs for a number of classes.

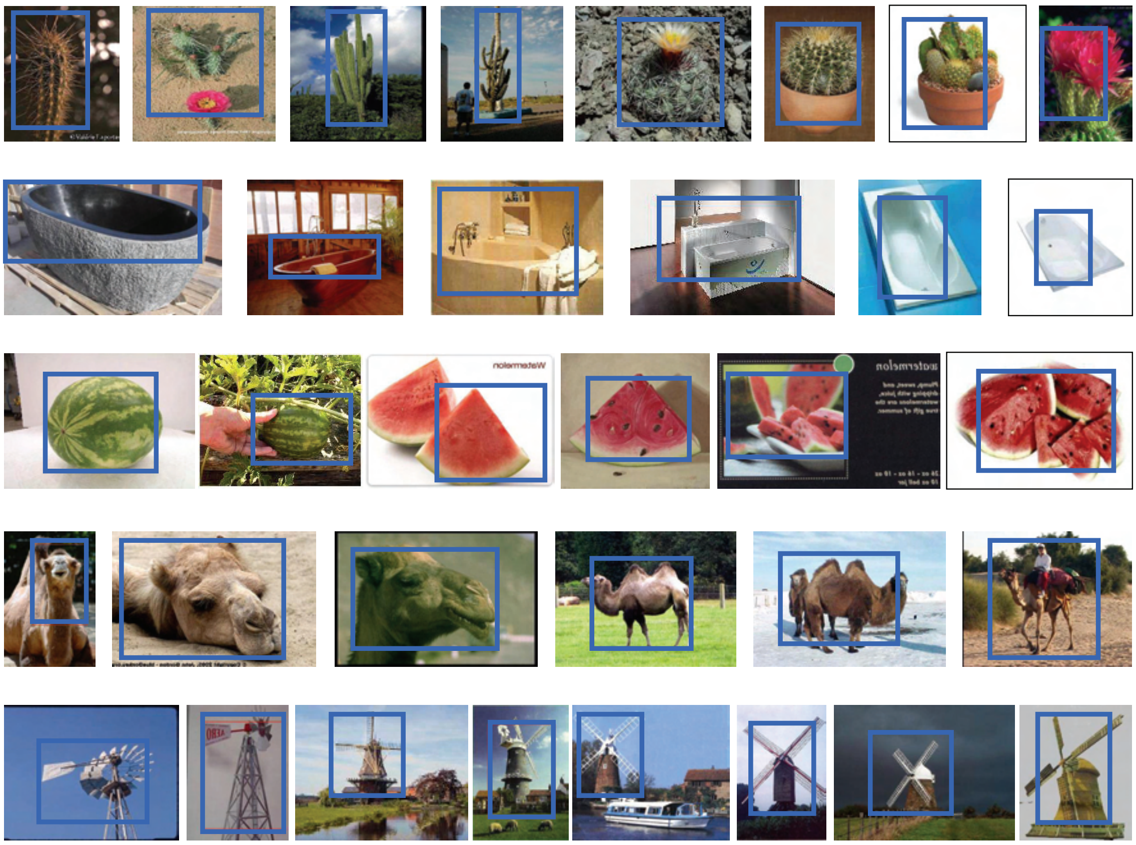


Figure 7.2: Automatic ROI detection. Examples from Caltech-256 for $s = 3$ for cactus, bathtub, watermelon, camel and windmill.

7.3 Random forests classifier

A random forest multi-way classifier consists of a number of trees, with each tree grown using some form of randomization. The leaf nodes of each tree are labelled by estimates of the posterior distribution over the image classes. Each internal node contains a test that best splits the space of data to be classified. An image is then classified by sending it down every tree and aggregating the reached leaf distributions. Randomness can be injected at two points during training: in subsampling the training data so that each tree is grown using a different subset; and in selecting the node tests.

Growing the trees

The trees here are binary and are constructed in a top-down manner. The binary test at each node can be chosen in one of two ways: (i) randomly, i.e. data independent; or (ii) by a greedy algorithm which picks the test that best separates the given training examples. "Best" here is measured by the information gain

$$\Delta E = -\Sigma_i \frac{\mid S_i \mid}{\mid S \mid} E(S_i) \tag{7.2}$$

caused by partitioning the set S of examples into two subsets S_i according the given test. Here $E(s)$ is the entropy $-\sum_{j=1}^{N} p_j log_2(p_j)$ with p_j the proportion of examples in s belonging to class j, N the number of classes, and $\mid . \mid$ the size of the set. The process of selecting a test is repeated for each nonterminal node, using only the training examples falling in that node. The recursion is stopped when the node receives too few examples, or when it reaches a given depth.

Learning posteriors

Suppose that T is the set of all trees, C is the set of all classes and L is the set of all leaves for a given tree. During the training stage the posterior probabilities ($P_{t,l}(Y(I) = c)$) for each class $c \in C$ at each leaf node $l \in L$, are found for each tree $t \in T$. These probabilities are calculated as the ratio of the number of images I of class c that reach l to the total number of images that reach l. Y(I) is the class-label c for image I.

Classification

Figure 7.3 shows a schematic example of classifying with a random forest. The test image is passed down each random tree until it reaches a leaf node. All the posterior probabilities are then averaged and the $\arg\max$ is taken as the classification of the input image.

7.3.1 Node tests for PHOG and PHOW

Recent implementations of random forests [78, 153] have used quite simple pixel level tests at the nodes (for reasons of speed). Here we want to design a test that is suitable

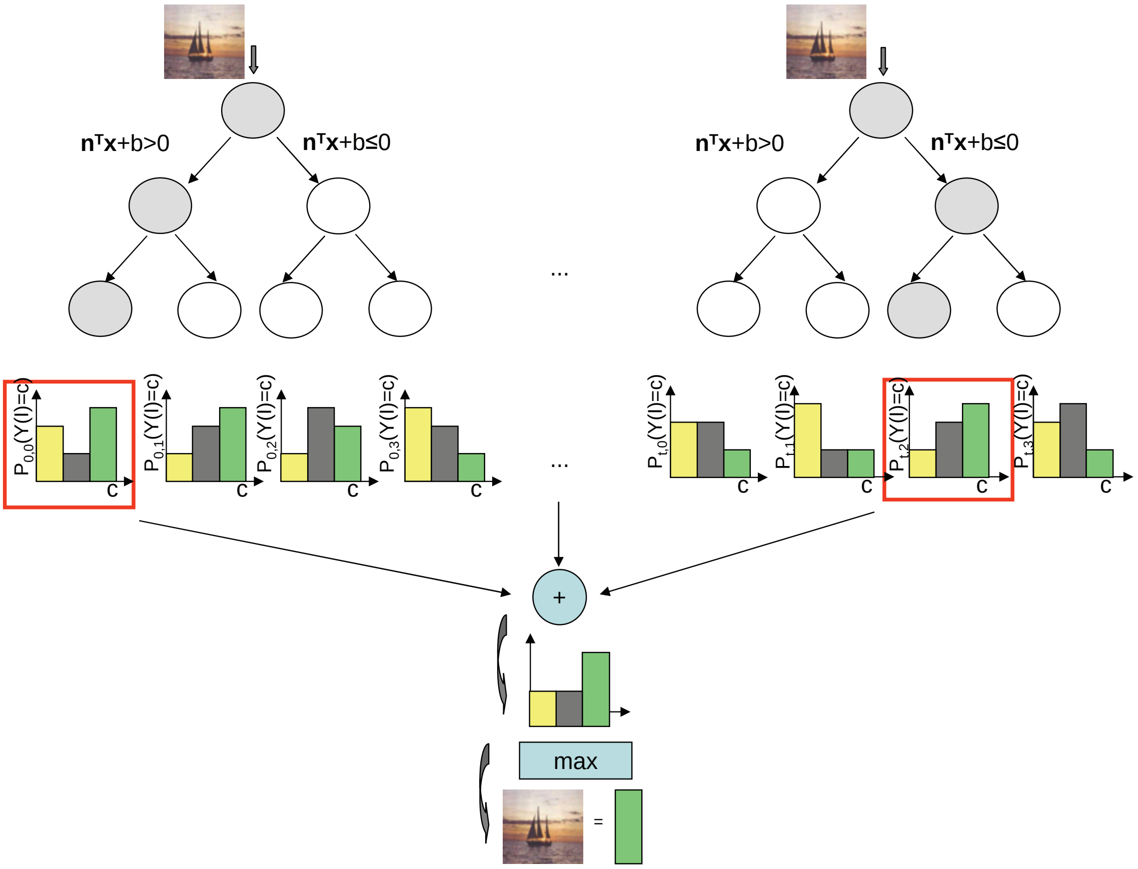


Figure 7.3: Schematic example of classification for Random Forests

for the representations of shape, appearance and pyramid spatial correspondence, that we have at our disposal.

We also use a relatively simple test – a linear classifier on the feature vector – but also include feature selection at the test level. The tests are represented as:

$$T = \begin{cases} if\ \mathbf{n}^{\mathbf{T}}\mathbf{x} + b \leq 0 & \text{go to the right child} \\ otherwise & \text{go to the left child} \end{cases}$$

where $\mathbf{n}$ is a vector with the same dimension as the data vector $\mathbf{x}$. A node test is obtained by choosing a random number of features n_f, choosing a random n_f indexes, and filling those components of $\mathbf{n}$ with random numbers in the range $[-1, 1]$ (the remaining components are zero). The value of b is obtained as a random number as well. Although this is a simple linear classifier we will demonstrate in section 7.8 that it increases speed and obtains comparable performances when compared with an M-SVM.

Descriptor selection

We wish to enable the classifier to be selective for shape or appearance or pyramid level – since some classes may be better represented by each of these. For example, airplanes by their shape, tiger by its appearance, classes with high intra-class variation by lower pyramid levels, etc.

To combine the features, a test selects the descriptor (shape or appearance). This is the same as giving weight 1 to one descriptor and weight 0 to the others. In both cases only one descriptor is used. In a similar manner pyramid levels are selected. A test selects a level l, and only the indexes corresponding to this level are non-zero in $\mathbf{n}$.

An alternative way to merge the descriptors is to build a forest for each descriptor (eg. 50 trees using only shape information and 50 trees using only appearance information) and merge them (using the 100 trees) for the classification.

7.4 Random ferns classifier

To increase the speed of the random forests Ozuysal *et al.* [109] proposed the random *ferns* classifiers. In the case of ferns there are an ordered set of nodes, and the node test is applied to the whole training data set. In contrast, in random forests only the data that falls in a child node is taken into account in the test. As in random forests leaves store the posterior probabilities. At each node in the fern set, a test gives a binary result (which in our case is 0 if $\mathbf{n}^{\mathbf{T}}\mathbf{x}+b>0$ or 1 if $\mathbf{n}tr\mathbf{x}+b\leq 0$). The result of each test and the ordering on the set defines a binary code for accessing the leaf node. For example, imagine we have 2 nodes each one making a partition for the whole training data, and each provides a binary result 0 or 1 according to the node test. Combining the results we can have 2^2 combinations: 00, 01, 10 and 11, which correspond to 4 leaves, one for each combination. So if a fern has N nodes, it will have 2^N leaves. To know which leaf a training image reaches, the process is the following: imagine the first and the second tests give result 0, the combination of both is 00 which means that the training image has reached leaf 0. If the first test gives result 1 and second test gives result 1 the combination is 11 meaning the training image has reached leaf 3, and so on. The advantage of ferns over forests is that it is not necessary to store the intermediate nodes (of a tree). As will be demonstrated in section 7.8, ferns improve on training speed, obtaining comparable results.

Figure 7.4 shows an example of randomized ferns. Again the test image is passed down all the randomized ferns. Each node in the fern provides a result for the binary test which is used to access the leaf which contains the posterior probability. The posteriors are combined over the ferns in the same way as for random forests over trees.

7.5 Image Classification

For the test images a "sliding window" over a range of translations and scales is applied. A new sub-image W_I classified by considering the average of the probabilities $P_{t,l}(Y(I) = c)$:

$$\hat{Y}(I) = \arg\max_c \frac{1}{T} \Sigma_{t=1}^{T} P_{t,l}(Y(I) = c) \tag{7.3}$$

where l is the leaf reached by image I in tree t. We classify an image I as the class C_k provided by the ROI which gives highest probability.

7.6 Datasets and methodology

Datasets

We provide results for object classification on Caltech-101 and Caltech-256 datasets. See chapter 3 for a complete description.

Methodology

Following standard procedures, the Caltech-101 data is split into 30 training images (chosen randomly) per category and 50 for testing – disjoint from the training images. For Caltech-256, 30 images are used for training and 25 for testing. For a comparison

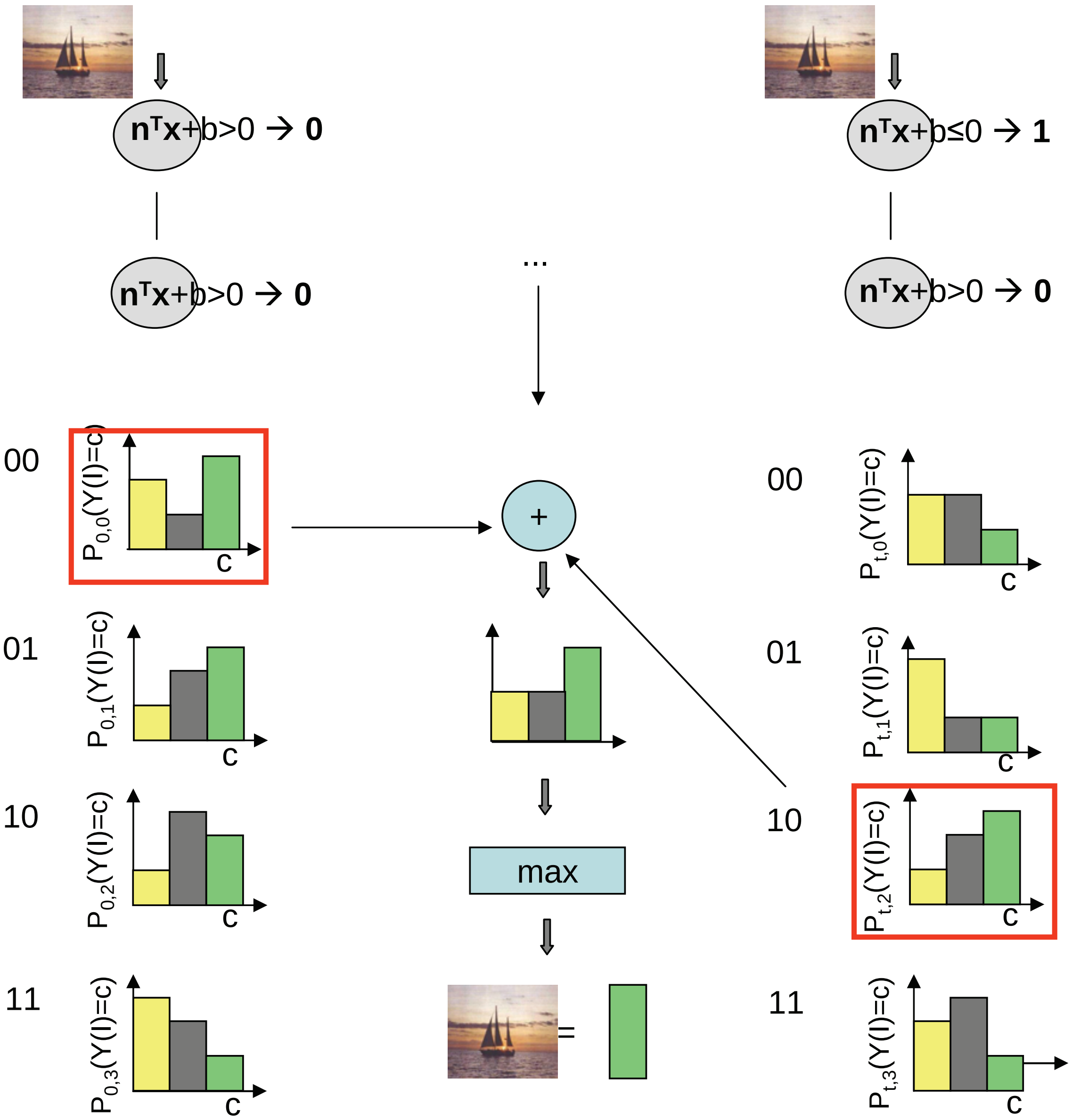


Figure 7.4: Schematic examples of classification for Random Ferns.

with [59] for Caltech-256, we report experiments without the last 6 categories and without clutter, this is 250 categories. The final performance score is computed as the mean recognition rate per class. The classification process is repeated 10 times, (changing the training and test sets), and the average performance score and its standard deviation are reported.

7.7 Implementation

Appearance, Shape and Pyramid level weighting

App_{Gray} and App_{Colour} are used for appearance representation and $Shape_{180}$ and $Shape_{360}$ are used for shape representation. We follow the same implementation procedure described in section 6.5.

The level weights α_l from (5.2) are learnt for each class separately by CLW optimization. Details are given in section 6.6.1.

ROI detection

The optimization process is done by testing the similarity between a different number of images s ranging from 1 to 4. The search is over the four parameters specifying the coordinates of the rectangle: x_{min}, x_{max}, y_{min} and y_{max}. The search is carried out over a translation grid with 10 pixel steps. We start the optimization process by fixing the four parameters to the image size and scaling them in steps of 0.1. At each iteration we optimize the cost function (7.1) for each training image. We stop when there are no more changes in the ROIs or when the number of iteration reaches 10. For the descriptor we use the PHOG and PHOW vectors concatenated. Both the level weights α_l, and a weight for each vector are obtained by optimizing the classification performance of that class against all others.

Randomized trees and ferns

At a given node, n_f features are randomly selected. The vector $\mathbf{n}$ is initialized with zeros and the n_f variables chosen are coefficients that are uniform random numbers on $[-1, 1]$. b is randomly chosen between 0 and the distance of the further point $\mathbf{x}$ from the origin. To inject randomness into the training set per tree, we randomly choose 10 training images per category, different for each tree, and the posterior probabilities in the terminal nodes are estimated from the remaining 20 training images. We then recursively build the trees by trying r different tests at each node and keeping the best one according to the criterion of (7.2). As in [78], for the root node we chose $r = 10$, a very small number, to reduce the correlation between the resulting trees. For all other nodes, we used $r = 100$D, where D is the depth of the node. This heuristic involves randomizing over both tests and training data. We do it to make our greedy algorithm tractable. When using the simpler approach, we grow the trees by randomly selecting $\mathbf{n}$ and b without measuring the gain of each test. For the two methods, trees are grown until a maximal depth is reached or until less than 10 instances fall in the node. We test trees for D = 10, 15 and 20. To grow the ferns $r = 10$ is used for each node.

7.8 Image classification results – Caltech-256

We first study the influence of different parameters using Caltech-256 as our test set, and then in section 7.9 we provide results for Caltech-101 as well as a comparison with the state-of-art. For the experiments the following parameters are used unless stated otherwise: 100 randomized trees with D=20, entropy optimization, and all the descriptors. Parameter optimization is carried out on a validation set (a sub-set of the training set, disjoint from the test set).

ROI

Table 7.1 shows the performances using all descriptors when changing the number s of images to optimize in the cost function. Without the optimization process the performance is 38.7%, and with the optimization this increases by 5%. There is not much difference between using 1 to 4 images to compute the similarity. Note, that when no ROI optimiza-

no optimization.	$s = 1$	$s = 2$	$s = 3$	$s = 4$
38.7 ± 1.3	42.5 ± 1.0	42.9 ± 1.0	43.5 ± 1.1	42.8 ± 1.0

Table 7.1: Caltech-256 performances when using 100 randomized trees with D=20, entropy optimization and all descriptors. First column is without ROI optimization and the rest are for ROI optimization from $s = 1$ to 4 images.

tion is used (as in previous chapter) very similar results are obtained with random forests (38.7%) and the M-SVM (40.1%) introduced in chapter 6.

Figure 7.2 shows some images with the detected ROI superimposed. Note that in all of them the algorithm is able to find the object in the images allowing us to have a more accurate image representation. However, in some cases the ROI overfits the object. If we have a look to the 4th and 5th camel images the ROI is smaller than the object, and we miss the head of the *camel*. This is because in the first image the camel is looking into the left side while int he second image the camel is looking into the right side. When computing the similarity the ROIs without the camel head are more similar than the ROIs including the camel head. Something similar happens with the 6th and 8th *windmill* images. If we have a look to the *bathtub* images, we are also missing some parts of the object. This can be due because this category has higher intra-class variation. May be for these categories (with higher intra-class variation) we need a smaller subset s and for others a bigger subset is better (e.g. *cactus*).

Node tests

The first two rows in table 7.2 compare the performances achieved using a random forests classifier with random node test (first row) and with entropy optimization (second row) when using $Shape_{180}$, $Shape_{360}$, App_{Colour}, App_{Gray} and when merging them. Slightly better results are obtained for the entropy optimization (around 1.5%). When merging all the descriptors with entropy optimization performance for random forests is 43.5%. Very similar results are obtained for appearance and shape descriptors. Taking the tests at random usually results in a small loss of reliability but considerably reduces the learning time. The time required to grow the trees drops from 20 hours for entropy optimization to 7 hours for random tests on a 1.7 GHz machine and Matlab implementation. First column

	Randomized Forests				
	Shp_{180}	Shp_{360}	App_C	App_G	All
RT	38.5 ± 0.8	39.3 ± 0.9	35.2 ± 0.9	39.3 ± 1.0	41.9 ± 1.2
EO	39.2 ± 0.8	40.5 ± 0.9	36.5 ± 0.8	40.7 ± 0.9	43.5 ± 1.1
	Randomized Ferns				
RT	37.7 ± 0.8	38.1 ± 0.8	34.7 ± 0.9	38.9 ± 0.9	41.0 ± 0.9
EO	38.9 ± 0.8	39.7 ± 0.9	36.5 ± 0.9	39.2 ± 0.8	42.6 ± 1.0

Table 7.2: Caltech-256 performance using appearance, shape and all the feature descriptors with randomized trees and ferns. 100 trees/ferns with D=20 and ROI with $s = 3$ are used. RT = Random Test, EO = Entropy Optimization.

in figure 7.5 shows how the classification performance grows with the number of trees when using all the descriptors (first row), when using shape descriptors (second row) and when using appearance descriptors (third row) with entropy optimization.

Forests vs ferns

The last two rows in table 7.2 are performances when using random ferns. Performances are less than 1% worse than those of random forests (42.6% when using all the descriptors to grow the ferns). The main advantage of using ferns is that the training time increases linearly with the depth, while for random forests it increases exponentially. For both, random forests and ferns the test time increases linearly with the number of trees/ferns.

For random forests we need to store $\sum_{d=1}^{D} 2^d$ nodes and for ferns we need to store $2^D + D$ nodes, so the size in memory to store a tree/fern increases exponentially with the depth D. The memory to store forests increases linearly with the number of trees/ferns.

Number of trees/ferns and their depth

Second column in figure 7.5 shows performances, as the number of trees increases, when varying the depth of trees from 10 to 20. First row is when merging all the descriptors, second row is for shape descriptor and third row for appearance. When merging all the descriptors performance is 32.4% for D=10, 36.6% for D=15 and 43.5% for D=20. When

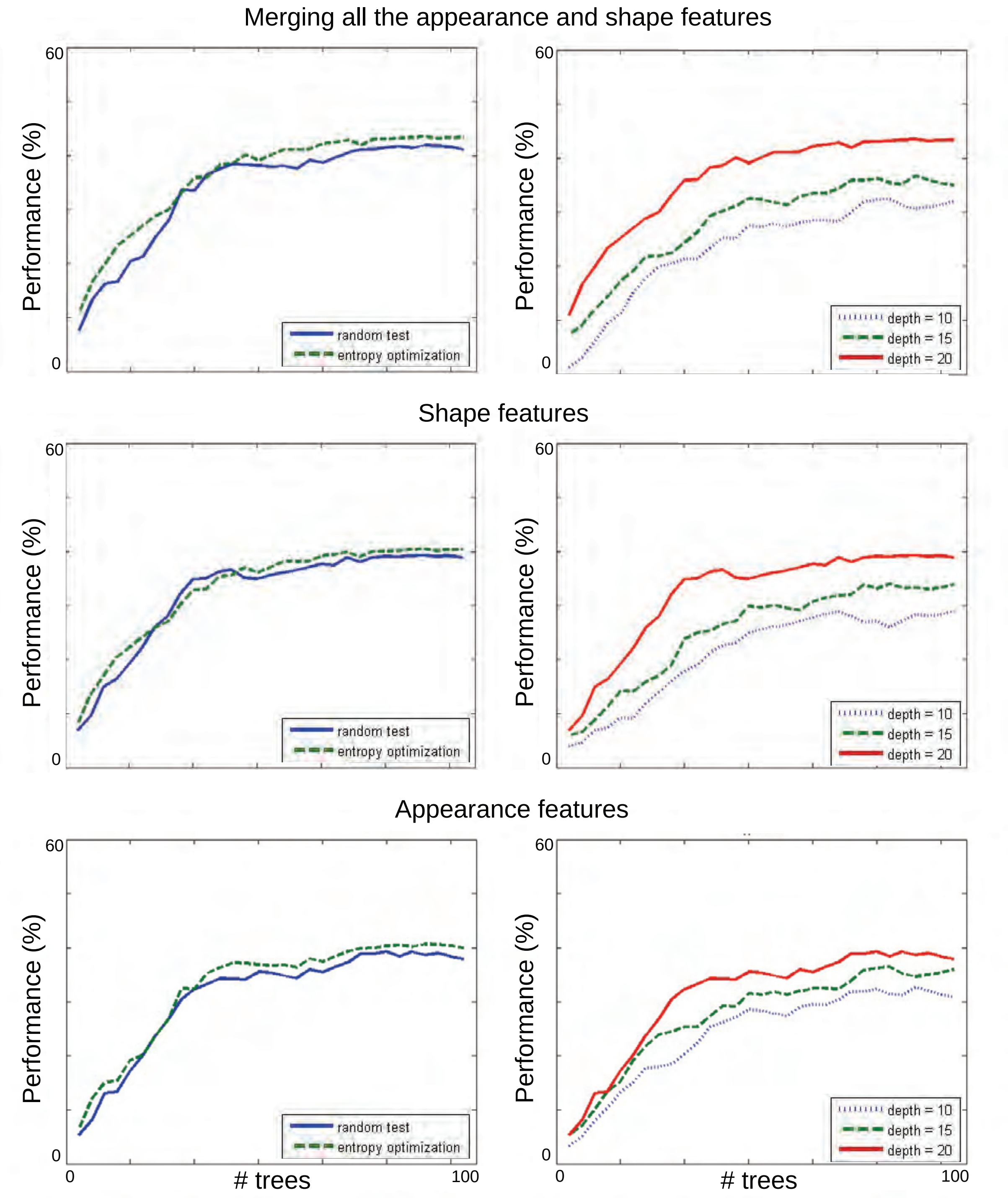


Figure 7.5: First column: Comparing the classification rates obtained using trees (D=20) grown by selecting tests that maximize the information gain (green line) and by randomly chosen tests (blue line), as a function of the number of trees. Note that when enough trees are used, the information gain does not improve the rates anymore. Second column: Comparing the classification rates obtained using trees with entropy optimization using D=10, 15 and 20 again as a function of the number of trees. All the descriptors are used in both graphs. First row: merging features; Second row: shape features; Third row: appearance features.

using ferns performances are 30.3%, 35.5 and 42.6 for D=10, 15 and 20 respectively. Increasing the depth increases performance however it also increase the memory required to store trees and ferns, as mentioned above.

When building the forests we experimented with assigning lower pyramid levels to higher nodes in the tree and higher pyramid levels to the bottom nodes. For example for a tree with D=20, pyramid level $l = 0$ is used until depth $d = 5$, $l = 1$ from $d = 6$ to $d = 10$, $l = 2$ from $d = 11$ to $d = 15$, and $l = 3$ for the rest. In this case performance decreases 0.2%. When merging forests by growing 25 trees for each descriptor (100 trees in total) and merging the probability distributions when classifying the performance increases 0.1%.

Number of features

All results above are when using a random number of features to fill the vector $\mathbf{n}$ in the linear classifier. Here we investigate how important the number of non-zero elements is. Figure 7.6a shows a graph, for both random and entropy tests, when increasing the number of features used to split at each node. The number of features is increased from 1 to m where m is the dimension of the vector descriptor $\mathbf{x}$ and $\mathbf{n}$. We can see that the procedure is not overly sensitive to the number of features used, as was also demonstrated in [23]. Very similar results are obtained using a single randomly chosen input variable to split on at each node, or using all the variables. This means that we can use a small number of features increasing the speed of the classifier in both training and testing.

Training data

Figure 7.6b shows how the performance changes when varying the number of training images to compute the posteriors. The training data to grow each tree is kept fixed to 10 and we increase the data to compute the posterior from 10 to 20 per image. Performance increases (by 1%) when using more training data meaning that with less data, the training set is not large enough to estimate the full posterior [5]. Since we do not have more than 30 images per category and, as noted in the graph, performance increases if we increase the training data, we populate the training set of positive samples by synthetically generating additional training examples [74]. Given the ROI, for each training

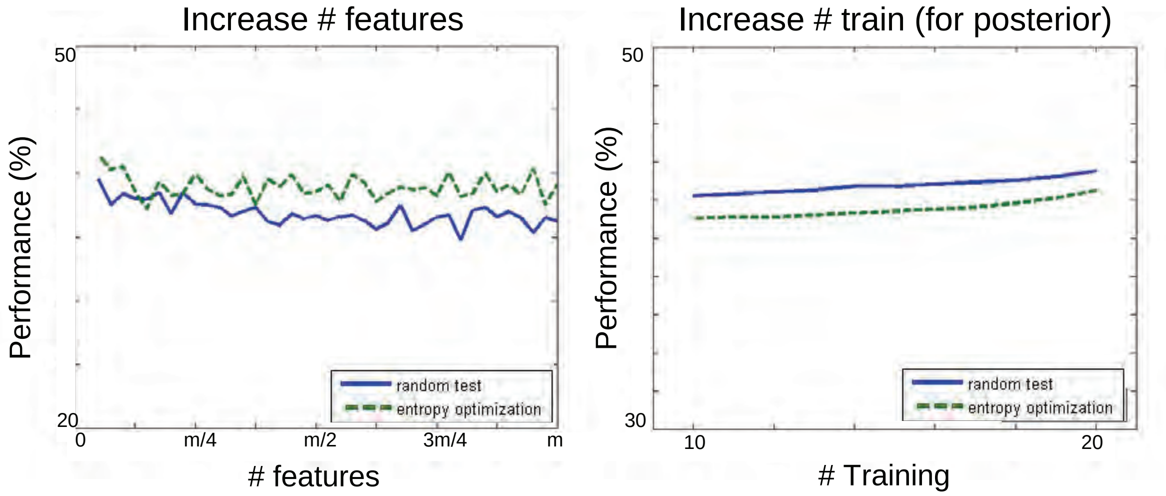


Figure 7.6: (a) performance when increasing the n_f to use at each node; (b) performance when increasing the number of images to compute the posterior probabilities. 100 trees with D=20, entropy optimization and all the descriptors are used in both graphs.

image we generate similar ROIs by perturbing the position ($[-20, 20]$ pixels in both x and y directions), the size of original ROIs (scale ranging between $[-0.2, 0.2]$) and the rotation ($[-5, 5]$ degrees). We treat the generated ROIs as new annotations and populate the training set of positive samples. We generate 10 new images for each original training example obtaining 300 additional images per category (resulting in a total of 330 per category) and the performance increases from 43.5% to 45.3%.

7.9 Comparison to previous results

In this section 100 trees with D=20 and entropy optimization to split each node is used. We use the ROI optimization and we increase the training data by generating 300 extra images per category. Table 7.3 summarizes the state of the art for Caltech-101 and Caltech-256.

7.9.1 Caltech-101

We first compare here our method with the one of Lazebnik *et al.* [76]. They use the pyramid kernel with appearance features and level weights $\alpha_0 = 0.25$, $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$ with an M-SVM. For Caltech-101 their performance is 64.6%. Random Forests

	[76]	[157]	[59]	Chap. 6	Ours
C-101	64.6	66.0	67.6	77.8	80.0 ± 0.6
C-256	–	–	34.1	40.1	45.3 ± 0.8

Table 7.3: Caltech-101 and Caltech-256 performances.

with the same PHOW descriptor improves to 73.7%.

A better comparison including the merging features performance can be obtained with the method introduced in chapter 6. In this case PHOW and PHOG are the feature vectors for an M-SVM classifier, using the spatial pyramid kernel (5.2). For merging features the kernel in (6.2) is used. Both class specific optimizations (CLW & CFW) are used here for comparison. Previous performance is 77.8%. Using this same features with random forests increases performance to 80.0%.

Compared to [157] (best state-of-art performance) our performance increases by 13%.

7.9.2 Caltech-256

We first compare our method with Griffin *et al.* [59]. Their performance is 34.1% using the PHOW descriptors with a pyramid kernel (5.2) and M-SVM. Random forests with the same PHOW descriptor improves to 40.7%. Our performance when merging different descriptors with random forests is 45.3%, outperforming the state-of-art by 11% and outperforming results in previous chapter by 5%. All the above results are for 250 categories. For the whole Caltech-256 (not clutter) performance is 44.0%.

Figure 7.7 shows our results and the results obtained by other authors for Caltech-101 and Caltech-256. Following the standard procedures of Caltech datasets we randomly select N_{train} = 5, 10, 15, 20, 25, 30, 40. $N_{test} = 50$ and $N_{test} = 25$ is used for Caltech-101 and Caltech-256 respectively. Note that for Caltech-101 adding the ROI optimization does not increase performance as much as in the case of Caltech-256. This is because Caltech-101 has less pose variation within the training images for a class.

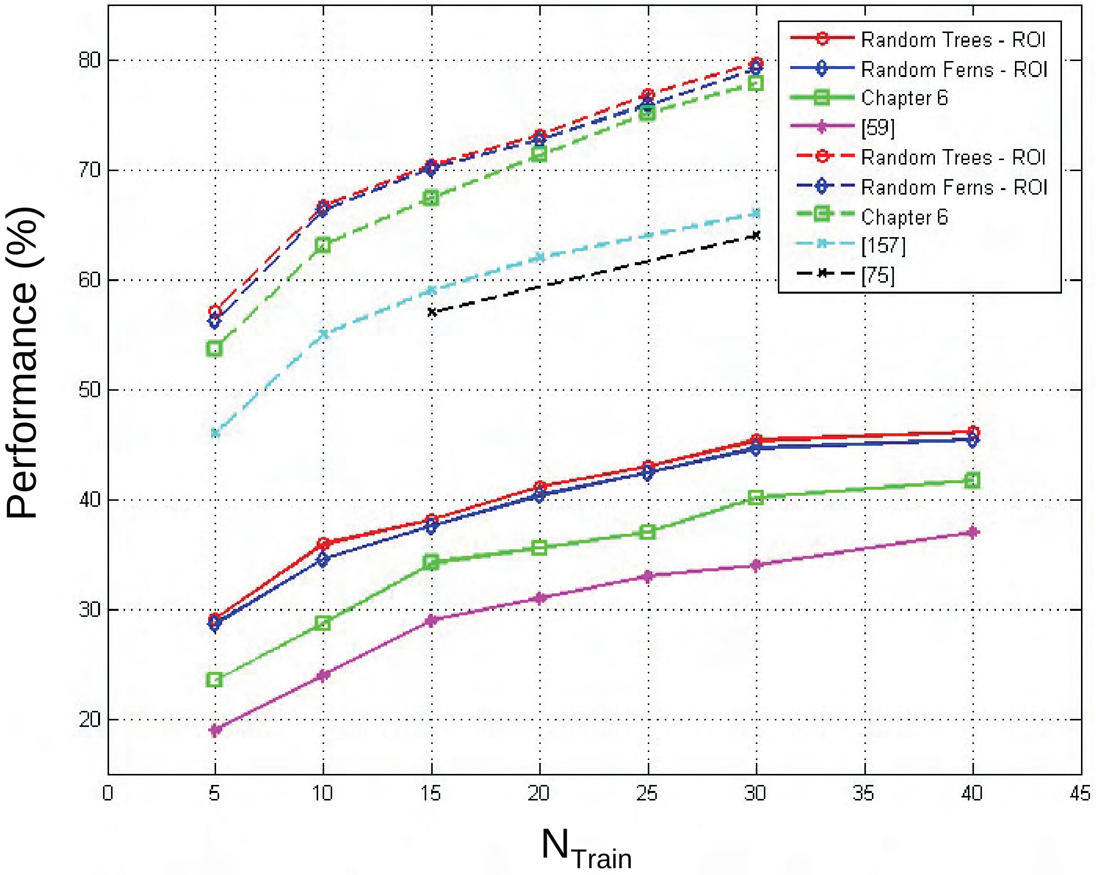


Figure 7.7: Performance as a function of the number of training images for Caltech-101 and Caltech-256. We compare our implementation with Random Forests/Ferns with the published results.

7.10 Summary

We have demonstrated that using random forests/ferns with an appropriate node test increases performance and speed over a multi-way SVM, and we have improved on the state of the art for Caltech-101 and Caltech-256.

In summary, we can approximately quantify the contributions arising from each of the principal improvements (summarized in figure 7.8): (i) Using the ROI detection and sliding window is a significant benefit. It increases performance from 2% to 5% depending on the degree of object pose variation within the datasets. (ii) Generating extra data during training increases performance by 2%. (iii) Using random forests/ferns instead of M-SVM, is less computational expensive, and comparable results are obtained (M-SVM

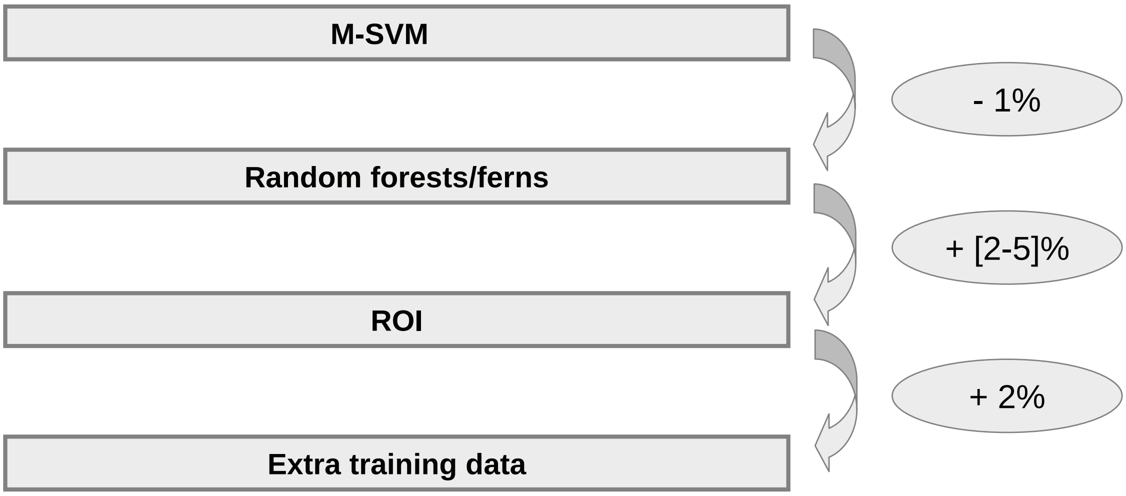


Figure 7.8: This graphs shows how much each representational and feature increment contributes to the overall performance.

is 1% better than Random Forests). More concretely, if using 25 test images per category for Caltech-256 dataset, it takes 60 hours to classify them when an M-SVM is used, and around 2 hours when a Random Forest is used.

Chapter 8

Conclusions

In this chapter we briefly summarize the main contributions of this thesis research and discuss some further work. Finally, publications which are directly related to this thesis are listed.

8.1 Summary and contributions of the thesis

In this thesis, we have proposed models and methods for image classification. In particular, we focused on classifying an image by the scene it belongs (e.g. coast, forest, living room, etc.) and classifying an image by the object it contains (e.g. sail boat, dolphin, cactus). The major contributions together with the improvement in performance and the conference/journal publications are summarized in figure 8.1 and explained below:

- An hybrid generative/discriminative model is presented in chapter 4 and published to the ECCV'06. It is based on discover the "objects" in the images by a generative learning process (pLSA) and further use these object distribution to classify the images as an scene category by using a discriminative classifier (SVM). We proposed to represent the image by using **colour sift** (appearance) descriptors on a regular grid and we showed that for natural scene classification this representation increases performance by 2% compared to SIFT features over gray level images.
- We included spatial information (PHOW) to the hybrid system in chapter 4. The hybrid system together with the spatial information extension has been published

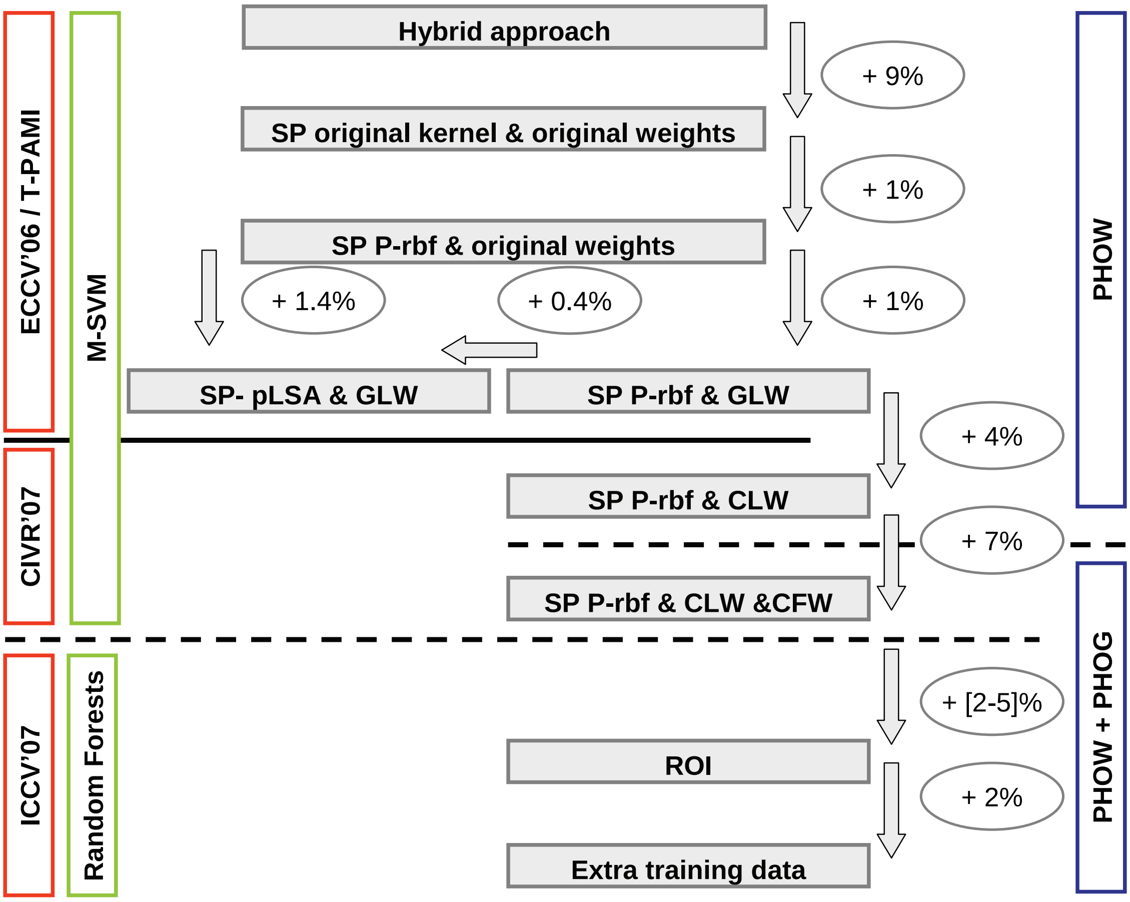


Figure 8.1: This graphs shows how much each representational and feature increment contributes to the overall performance on Caltech-101 and Caltech-256.

to the T-PAMI journal. We introduced a **P-rbf kernel** which increases performance by 1% compared to the SP kernel proposed in [76]. Moreover we generalized the original kernel and *learn* the level weights (**GLW optimization** on a validation set), increasing performance a further 1%. We successfully incorporated the spatial information (PHOW) to the hybrid system (**SP-pLSA**) and show that it increases performance by 2.4% respect to the method proposed by Lazebnik *et al.* [76].

- A new shape descriptor that supports spatial pyramid matching (**PHOG**) is presented in chapter 6 and published to CIVR'07. It generalizes the PHOW representation of Lazebnik *et al* [76] from appearance (visual words) alone to local shape (edge distributions). We demonstrated that this descriptor performs as well as local appearance patches. Moreover we learn a class specific weighting for the levels of the hierarchical spatial histogram (**CLW optimization** on a validation set) showing that low levels are better for objects with higher intra-class variation. This learning increases performance by 4%. Finally we demonstrate that shape and appearance kernels may be combined. We showed that for some categories shape works better than appearance, while for some others appearance is better, and in some cases a combination of both is more appropriate. We introduce a **kernel for merging** features and again the best features parameters are learnt on a validation set (**CFW optimization**). The merging features with parameter learning increases performance by 7%.

- A random forest/fern classifier which combines appearance and shape features over a ROI is proposed in chapter 7 and submitted to ICCV'07. We show that using random forests similar performances to an M-SVM are obtained. We propose a method for **automatically select** the regions of interest (**ROI**) in training. This provides a method of inhibiting background clutter and adding invariance to the object instance's position. The PHOW and PHOG over the ROI are used to represent the images. Using a ROI increases performance by $2-5\%$ and if we generate **extra training data** we increases performance another 2%. We introduce a **node test** which is able to work with different pyramid level representations as well as different cues (appearance and shape).

All theses contributions achieve superior classification accuracy compared to recent publications, in all cases using the authors' own datasets and testing protocols. Specifically for Caltech-101 and Caltech-256 we outperform the state-of-art by 14% and 11% respectively.

8.2 Discussion

Even we have achieved the highest performances for the Caltech-101 (80%) and Caltech-256 (45%) datasets, we are still far from the 100% of correct classification rates. If we have a look at the most confused categories we can see that most of them are very related categories. For example the *schooner* and *ketch* images shown in figure 8.2 are very related. People who are not into sailing would classify both categories as a *sail boat*

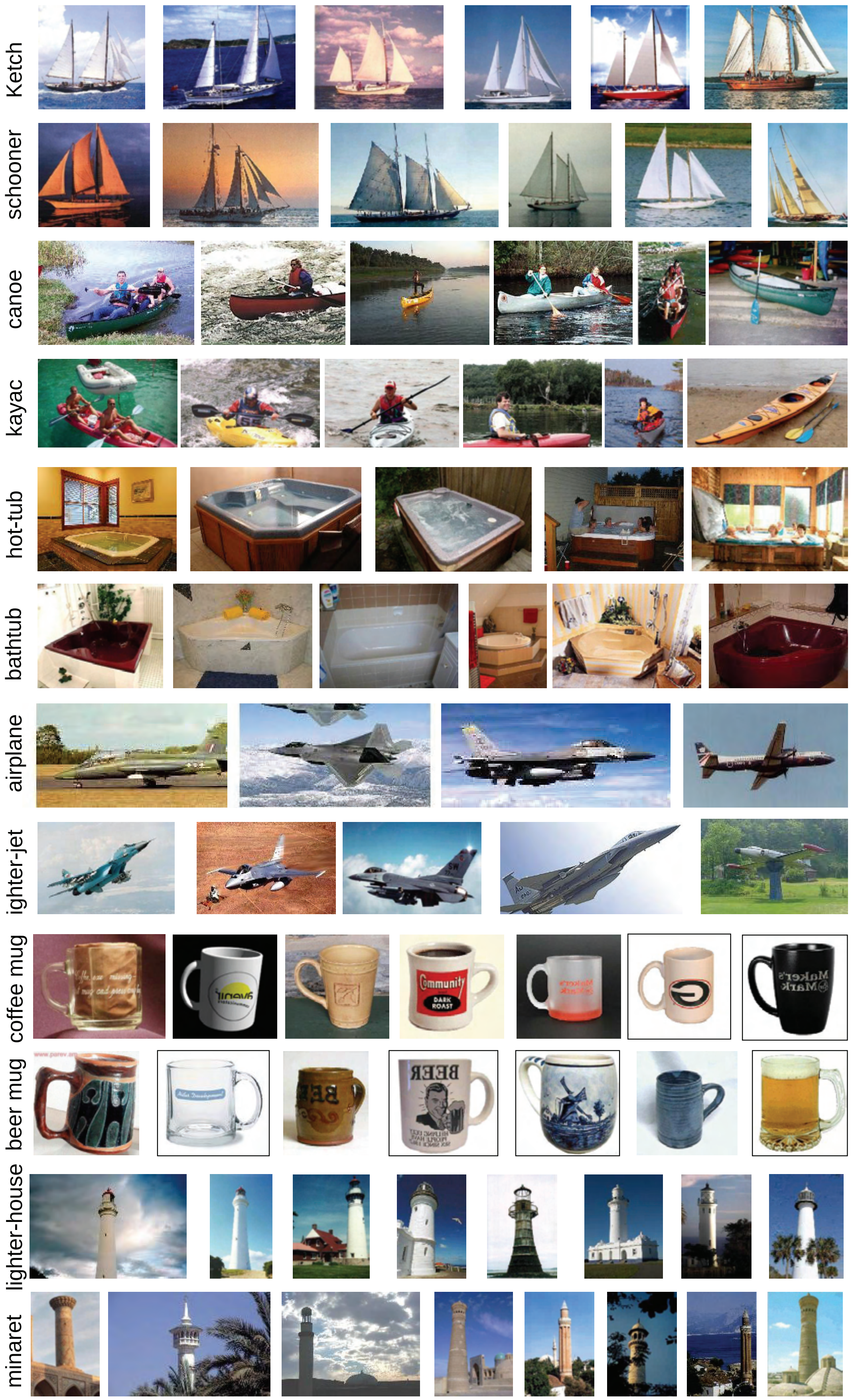


Figure 8.2: These images are related confused categories. Note that most of the humans, if they are not expert, would confuse them as well.

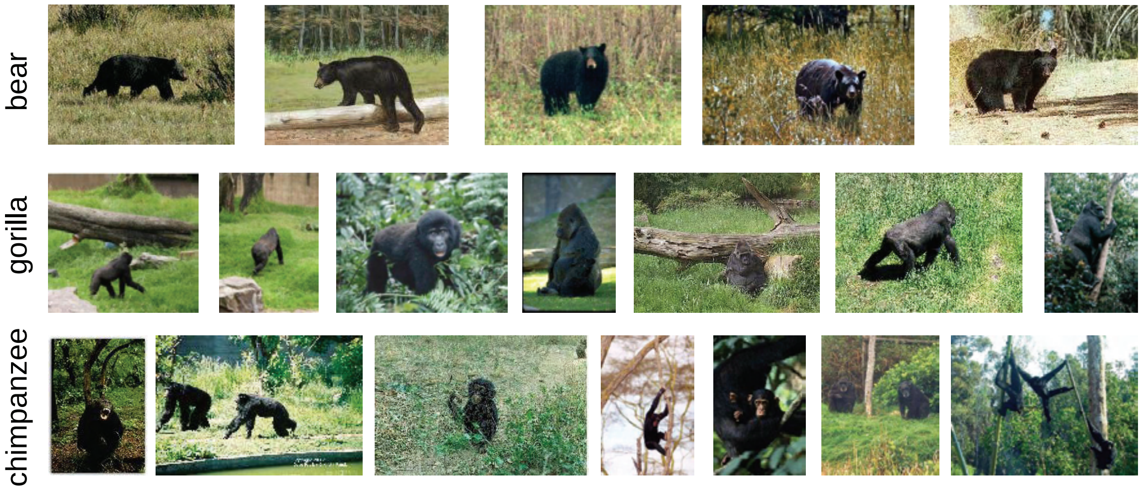


Figure 8.3: These images are categories which contain very “similar” animals.

because they are not able to distinguish between them. This same happens with the *canoe* and *kayak*, *hot-tub* and *bathtub*, *airplane* and *fighter-jet*, etc. categories in figure 8.2, all of them can be easily confused even by humans. Other related confused categories are the ones shown in figure 5.11 and figure 6.7.

Let’s have a look now at figure 8.3. The first row correspond to a *bear* category, the second row correspond to *gorilla* and the third to *chimpanzee*. These ones are also images confused by the system, however in this case humans are able to recognize them, why? We know that bears can not climb up the trees, we know that gorillas and chimpanzees front legs are larger than the bear front legs, we know that these three animals have different “faces”, however our system is not able to think about all these. Even though, note that chimpanzees and gorillas can sometimes be confused even by humans.

These confusions are not ambiguities like in the scene classification, they are due to the big similarity between some categories. In the scene classification it is not worth to try a hard assignment between ambiguous categories, however for object classification a hard assignment is possible. Humans in general are not able to distinguish between related categories in figure 8.2, so a general system could group these categories into a single one. However expert humans in the field could classify and distinguish theses related images. A solution would be to design an “expert system” able to distinguish amongst only these related categories.

8.3 Further work

There are several ways further research could go, but there are also a variety of obvious extensions to the existing frameworks we have presented here in this thesis. We following briefly describe them:

- **Vocabulary generalization**. We have done some preliminary experiments about vocabulary generalization (see section 4.8). However the two datasets used have very similar image categories. A further work can be to explore how the vocabulary can be generalized if using datasets with different categories. We can build a universal vocabulary with better generalization to new unseen objects.
- **Texture**. We have explored appearance and shape features. A further work would be to explore how the texture information can help in classification.
- **ROI detection**. As we discussed in section 7.8 it seems that classes with higher intra-class variability can work better with lower values of s (number of images that have "corresponding" object instances). We can improve the ROI detection by choosing a class-specific subset s of images for each category instead of using the same number s for all the categories.
- **Multiple ROI detection**. Another thing to bear in mind is the number of object instances that appear in the images. At the moment only a single object instance can be detected, a further work would be to automatically detect the ROIs for all the object instances in the images.
- **Feature weights in random forests**. At the moment at each node test only one feature (shape or appearance) is used. This provides a good representation because for example *leopards* are best distinguished by their appearance while *cars* are best distinguished by their shape. However there is other categories for which both cues are useful (e.g. *Buddha*) and currently we are not taking this into account. We could do the node test by giving a weight to the features vectors (as we did for CFW optimization) and consequently use all the information we have. For example we can choose random weights ranging from 0 to 1 to weight the vector descriptors. If the weight is 0 or 1 we have the current representation, however if the weight is a

value between 0 and 1, let's say 0.3 for shape and 0.7 for appearance, we will take into account both descriptors. The same happens with the pyramid level. At the moment we are only using one pyramid level, we could improve it by giving some weights at each level.

- **Invariances**. We have shown that the ROI provides some translation and scale invariances. However due to the nature of the spatial pyramid matching the question about the rotation invariance is still open. It would be nice to design a system to deal with rotated object instances.
- **Hierarchical model**. We have also addressed the question about close related categories. These categories (e.g. ketch and schooner) are the most confused categories by our system and also by the humans if we are not experts on the topic. We can extend our system to automatically learnt a category hierarchy. The highest levels will be able to classify amongst the easiest separable categories, grouping the related ones as the same category. For example the ketch and schooner category will be both classified as a sail boat. Then at lower levels we can try to find more specific features to distinguish amongst these more related categories.
- **Unknown categories**. The current system is unable to recognize categories which have not been considered in the learning stage. For example if we train the system to recognize *cars* and *motorbikes*, and the test image contains a *bicycle* it would be wrongly classified as *car* or *motorbike*. However, the ideal solution would be a system able to say that it does not know the category in the test image.
- **CBIR**. Image categorization is a special case of image retrieval where the query corresponds to the image category being searched for [150]. We have demonstrated that we can build a good image representation so our method could easily be used for image retrieval tasks providing an automatic ranking of scenes.

8.4 Related publications

The following publications are a direct consequence of the research carried out during the elaboration of the thesis, and give an idea of the progression that has been achieved. Publications in both fields Image Classification and Medical Image are provided.

Journals

- A. Bosch, A. Zisserman, X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (T-PAMI).

- A. Bosch, X. Muñoz, R. Martí. A Review: Which is the best way to organize/classify image by content?. *Image and Vision Computing* (IVC). Volume 25, Issue 6 , Pages 778-791, June 2007.

- A. Bosch, X. Muñoz, J. Freixenet. Segmentation and description of natural outdoor scenes. *Image and Vision Computing* (IVC). Volume 25, Issue 5, Pages 727-740, May 2007.

International conferences

- A. Bosch, A. Zisserman, X. Muñoz. Image Classification using Random Forests and Ferns. *International Conference on Computer Vision* (ICCV 2007). Submitted

- A. Bosch, A. Zisserman, X. Muñoz. Representing shape with a spatial pyramid kernel.*ACM International Conference on Image and Video Retrieval* (CIVR 2007). To appear

- J. Philbin, A. Bosch, O. Chum, J. Geusebroek, J. Sivic, A. Zisserman. Oxford TRECVID 2006 - Notebook Paper. *Proceedings of the TRECVID 2006.*Workshop (2006).

- A. Bosch, X. Muñoz, A. Oliver, R. Martí. Object and Scene classification: what does a Supervised Approach Provide us?. *International Conference on Pattern Recognition* (ICPR 2006). Vol I, pp 773-777. Hong Kong. August 2006.

- A. Oliver, J. Freixenet, R. Martí, A. Bosch, J. Martí. A New Approach to the Classification of Mammographic Masses and Normal Breast Tissue. *International Conference on Pattern Recognition* (ICPR 2006). Vol IV, pp 707-710. Hong Kong. August 2006.

- A. Bosch, X. Muñoz, A. Oliver, J. Martí. Modelling and Classifying Breast Tissue Density in Mammograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR 2006). Vol II, pp 1552-1558. New York. June 2006.

- A. Bosch, A. Zisserman, X. Muñoz. Scene Classification via pLSA. *9th European Conference on Computer Vision* (ECCV 2006). Vol IV, pp 517-530. Graz, Austria. May 2006.

- A. Bosch, X. Muñoz, J. Martí. Using appearance and context for outdoor scene object classification. *International Conference on Image Processing* (ICIP 2005). Vol II, pp 1218-1221. Genova, Italy. September 2005.

- A. Bosch, X. Muñoz. J. Freixenet, J. Martí. Supervised Object Knowledge learning for Image Understanding. *Proceedings of the Quality, Automation and Robotics Conference* (Q&A-R 2004). Cluj-Napoca, Romania. May 2004.

National conferences

- X. Muñoz, A. Bosch, J. Martí, J. Espunya. A Learning Framework for Object Recognition on Image Understanding. *Iberian Conference on Pattern Recognition and Image Analysis* (IbPRIA 2005). pp 311-318. Estoril, Portugal. June 2005.

- A. Oliver, J. Freixenet, A. Bosch, D. Raba, R. Zwiggelaar. Automatic Classification of Breast Tissue. *Iberian Conference on Pattern Recognition and Image Analysis* (IbPRIA 2005). pp 471-478. Estoril, Portugal. June 2005.

- A. Bosch, X. Muñoz, J. Martí, A. Oliver. Classifying Natural Objects on Outdoor Scenes. *Congrés Català d'Intel·ligència Artificial* (CCIA 2005). pp 115-122, L'Alguer, Italy. October 2005.

- J. Martí, J. Freixenet, D. Raba, A. Bosch, J. Pont, J. Español, y cols. HRIMAC una Herramienta de Recuperación de Imágenes Mamográficas por Análisis del Contenido para el Asesoramiento del Cáncer de Mama. *Informática de la salud*. Madrid, 2003.

Technical reports

- A. Bosch, X. Muñoz. Estudi de Diferents Descriptors de Forma per Caracteritzar Objectes, Technical Report IIiA 05-01-RR, Institute of Informatics and Applications. University of Girona, 2005. (Catalan)

Appendix A

Terminology and abbreviations

A.1 Terminology

Many terms related to image classification are used in a somewhat loose manner in the literature so to avoid confusion we give definitions of the terms used in the thesis:

- *Scene/Object annotation*: Also scene object labelling. Consists on manually annotate/label the images as a kind of scene or object it contains.

- *Image classification*: Is the task to classify an image as the scene/object it contains.

- *Image category*: Refers to the label for the whole image. Thus we can have coast image (image representing a coast scene) or a dolphin image (image which contains a dolphin).

- *Category*: Refers to a visually consistent set of scenes or objects.

- *Supervised learning*: A learning method is called supervised if the method needs the labels of the training images (what kind of scene) and the segmentation and localization of objects in images.

- *Weakly-supervised learning*: Weakly supervised means that no information about the object location in the training images is given. Hence, only the labels of the training images are provided.

- *Unsupervised learning*: An approach is said to learn unsupervised if it gets a pile of images for training without any hint about the image labels or the object location in those images.
- *Discriminative vs generative learning*: Generally the approaches for scene classification or object recognition can be split into two groups, discriminative approaches and generative ones. A discriminative approach learns a decision boundary that separates the two categories in feature space. That means the learning algorithm tries to directly model posterior probabilities. The generative approach tries to model the class-conditional densities and the priors for the class separately. The posterior probability is then evaluated from these two entities using Bayes Theorem.
- *Topic*: It has a similar meaning than object. The difference is that we know the object mountain, and this one can be divided in two topics: mountain with snow and mountain without snow. It is a term that we will use to denote the ”objects” that the system recognizes without supervision.
- *Visual word*: Is the analogy of the term word of the text analysis. It denotes specific informative parts of an image.
- *Visual Vocabulary*: It is composed for a set visual words.
- *Bag-of-words*: Sometimes called bag-of-features or bag-of-visterns. The image is represented as a ”bag” of representative features. Each image is further encoded by a binary vector whether it contains certain visual words or not. In a more general way it refers to the visual word histogram of an image.

A.2 Abbreviations

Here below we summarize the abbreviations used in the thesis.

- **CBIR**: Content Based Image Retrieval
- **CFW**: Class-specific Feature Weight optimization
- **CLW**: Class-specific Level Weight optimization

- **FP**: Fei-Fei and Perona [41] dataset
- **GFW**: Global Feature Weight optimization
- **GLW**: Global Level Weight optimization
- **KNN**: K-Nearest Neighbour
- **LSP**: Lazebnik *et al.* [76] dataset
- **OT**: Oliva and Torralba [102] dataset
- **PHOG**: Pyramid Histogram of Orientation Gradients
- **PHOW**: Pyramid Histogram Of visual Words
- **pLSA**: probabilistic Latent Semantic Analysis
- **P-rbf**: Pyramid-radial basis function kernel
- **RF**: Relevance Feedback
- **ROI**: Region Of Interest
- **SP**: Spatial Pyramid kernel
- **SP-pLSA**: Spatial Pyramid pLSA
- **SVM**: Support Vector Machine
- **VS**: Vogel and Schiele [149] dataset

Appendix B

A medical image application

In this appendix we present a new approach to model and classify breast parenchymal tissue by using the method introduced in chapter 4. Given a mammogram, first, we will discover the distribution of the different tissue densities in an unsupervised manner, and second, we will use this tissue distribution to perform the classification. We achieve this using a classifier based on local descriptors and probabilistic Latent Semantic Analysis (pLSA).

We studied the influence of different descriptors like texture and SIFT features at the classification stage showing that textons outperform SIFT in all cases. Moreover we demonstrate that pLSA automatically extracts meaningful latent aspects generating a compact tissue representation based on their densities, useful for discriminating on mammogram classification. We show the results of tissue classification over the MIAS and DDSM datasets. We compare our method with approaches that classified these same datasets showing a better performance of our proposal.

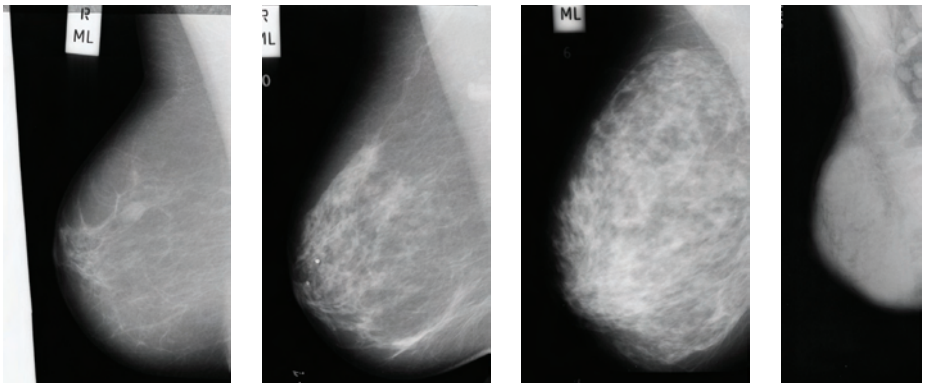
B.1 Why tissue classification?

Breast cancer is considered a major health problem in western countries. A recent study from the National Cancer Institute (NCI) estimates that, in the United States, about 1 in 10 women will develop breast cancer during their lifetime [2]. Moreover, in such country, breast cancer remains the leading cause of death for women in their 40s [26].

Although manual screening of mammographies remains the key screening tool for the detection of breast abnormalities, it is widely accepted that automated Computer Aided Diagnosis (CAD) systems are starting to play an important role in modern medical practices. Most of the commercially available CAD systems and research efforts in breast mammography focuses only on the automatic detection of abnormalities. However, from a medical point of view, it is well-known that there is a strong positive correlation between high breast parenchymal density and high breast cancer risk [155]. For instance, the relative risk is estimated to be about 4 to 6 times higher for women whose mammograms have parenchymal densities over 60% of the breast area, as compared to women with less than 5% of parenchymal densities [160]. Thus, the development of automatic methods for classification of breast tissue is justified for an automatic risk assessment framework in prospective CAD systems. However, developments in this area have been limited.

Several techniques have been proposed for breast density classification [69, 160], but only a small number of previous works have suggested that texture representation of the breast might play a significant role. Miller and Astley [95] investigated texture-based discrimination between fatty and dense breast types applying granulometric techniques and Laws texture masks. Byng *et al.* [27] used measures based on fractal dimension. The work of Bovis and Singh [22] first estimated features from the construction of Spatial Gray Level Dependency matrices and second, it trains multiple Neural Nets (ANN) to classify the parenchymal density. Zwiggelaar *et al.* [161] segmented mammograms into density regions based on a set of co-occurrence matrices, and density classification used the size of the density regions as the feature space. Similarly, Oliver *et al.* [104, 106] proposed to extract texture features after the segmentation of the breast in two clusters which represent dense and fatty tissue.

Motivated by the good results in image classification obtained by the hybrid model introduced in chapter 4, we propose to use it to classify the mammograms tissue. To carry out the adaption of this method to the medical image domain, we established the following analogies: in tissue classification, the *images* will be the *mammogram*, the *topics* will be the different *densities of the tissue* and we also will talk about *visual words* as the analogue of a *word*. pLSA is appropriate here because it provides the correct statistical model for clustering in the case of multiple tissues densities per image. We will have to study which are the best descriptors when classifying parenchymal densities as well as which is the best representation for this kind of images. Our main contribution in mammogram



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

Figure B.1: Four images belonging to one of each BI-RADS category extracted from MIAS dataset: from (a) BI-RADS I to (d) BI-RADS IV.

tissue classification is that this algorithm is able to learn relevant intermediate representation of tissue density automatically and without supervision. The previous approach of Petroudi *et al.* [113] which uses histogram models of textons does not provide a strong statistical model as our and can not differentiate the different densities in a mammogram automatically.

Nowadays, the American College of Radiology (ACR) Breast Imaging Reporting and Data System (BI-RADS) [3] is becoming a standard on the assessment of mammographic images. This standard provides four categories according to breast parenchymal density (see also figure B.1):

- BI-RADS I: the breast is almost entirely fatty.
- BI-RADS II: there is some fibrogandular tissue.
- BI-RADS III: the breast is heterogeneously dense.
- BI-RADS IV: the breast is extremely dense.

The rest of this annex is described below. Section B.2 presents a detailed overview of the proposed system. Sections B.2.1 and B.2.2 describe a previous segmentation step and how we are going to represent the image using local descriptors. Section B.2.3 briefly reviews the image representation and classification process, here used to classify the mammograms according their parenchymal density. Section B.3 describes the dataset and the followed methodology to test the approach. Section B.4 show the results obtained and a brief comparison. The paper ends with conclusions and outlines possible future work.

B.2 System overview

In this section we will explain the three processes involved on the mammographic tissue classification: (i) segmentation of the breast profile, (ii) breast tissue representation using a bag-of-words, and (iii) the use of probabilistic Latent Semantic Analysis (pLSA) to obtain the tissue classification according to the BI-RADS standard. Figure B.2 shows the schema of the system.

B.2.1 Pre-processing steps

The initial step of our approach is the segmentation of the profile of the breast. Previous works on breast tissue classification and abnormalities detection noticed that the feature extraction process is affected if the region processed is not well focused. Thereby, it is important to segment the mammogram in order to extract the breast from other objects that could be present in a mammographic image (background, annotations, pectoral muscle in MLO images) and to achieve optimal breast parenchyma measurements. We used a *two-phase* based method:

- **Breast Segmentation**. The algorithm computes a global gray histogram for the image. The gray values are represented by a histogram with 8 bins. We compute an automatic threshold which is the minimum value over the 8-histogram. This one is used to threshold the image obtaining a collection of different regions. The largest region (the union of the breast and the pectoral muscle) is extracted using a Connected Component Labeling algorithm. As a result we delete the labels and the information which is not necessary and we obtain an image with the segmented breast.

- **Pectoral Muscle Extraction**. This operation is important in mediolateral oblique view (MLO), where the pectoral muscle, slightly brighter compared to the rest of

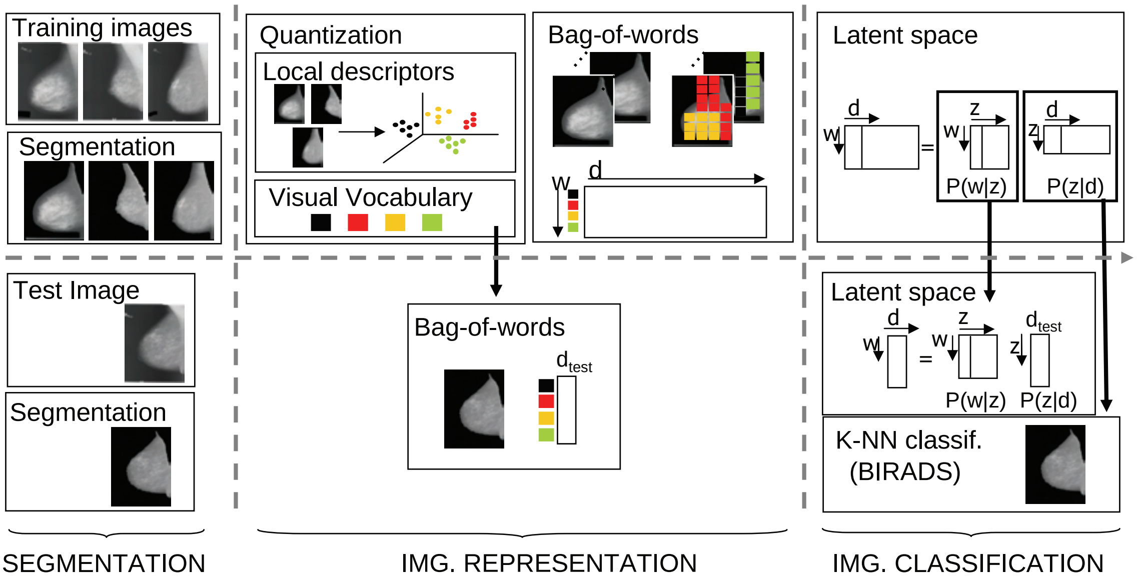


Figure B.2: Schema of the learning and classification process. The first row shows the learning process while the second one shows the classification process. The figure is divided in three subparts corresponding to the three main process of our approach: (a) segmentation; (b) image representation; and (c) use of the latent space for learning and image classification.

> the breast tissue, can appear in the mammogram. We used the approach of Ferrari *et al.* [46] who propose a polynomial modeling of the pectoral muscle.

This segmentation results in a minor loss of skin-line pixels in the breast area, but those pixels are not relevant for tissue estimation.

B.2.2 Image representation

We will represent the images as a co-occurence table (bag-of-words) built from automatically extracted and quantised descriptors. Given the set of training images, local descriptors are computed around the pixels of the tissue (we do not take into account points close to the border) and a vocabulary of visual words (visual vocabulary) is obtained. In order to obtain the visual vocabulary, we used two different measures: the first one based on the appearance (textons) and the second one based on the edge orientation (Scale Invariant Feature Transform - SIFT).

- **Textons**: As in [147], a $N \times N$ square neighbourhood is opened around each pixel. The pixels are row reordered to form a vector in an N^2 dimensional feature space. The patch size tested are N = 3, 5, 7, 11, 15 and 21. The patches are spaced by M pixels on a regular grid over the area of the tissue. The patches do not overlap when $M = N$, and do overlap when $M = 2$ (for N = 3, 5, 7) and M = 7 (for N = 11, 15 and 21).

- **SIFT**: SIFT descriptors [84] are computed at: (i) points on a regular grid with spacing M pixels, here M = 5 and 10. At each grid point SIFT descriptors are computed over circular support patches with radii r = 8 and 16 pixels; (ii) Affine co-variant regions are computed for each grey scale image, constructed by elliptical shape adaptation about an interest point [93]. Consequently each point is represented by a 128-dim SIFT descriptors. Note, the descriptors are rotation invariant.

The number of descriptors is around 35000 and depends on how big is the area of the tissue and the parameters N and M. The visual vocabulary (V) is obtained by vector quantising the descriptors computed from the training images using k-means.

Once we obtain the vocabulary, we represent the mammogram. Suppose we have a collection of images (mammograms) D = d_1,...,d_N with words from a visual vocabulary W = w_1,...,w_V. One may summarize the data in a V × N co-occurrence table of counts N_{ij} = $n(w_i, d_j)$, where $n(w_i, d_j,)$ denotes how often the term w_i occurred in an image d_j.

B.2.3 Image classification

Once we have built the bag-of-words we will use pLSA [65] to automatically find the topic (tissue) distribution for each mammogram. These distributions will be further used by the K-Nearest Neighbour (K-NN) or Support Vector Machines (SVM) to perform the mammogram classification as it is shown in figure B.2c.

In training stage, the topic specific distributions $P(w|z)$ are learnt from the set of training images. Each training image is then represented by a Z-vector $P(z|d_{train})$, where Z is the number of topics learnt. Determining both $P(w|z)$ and $P(z|d_{train})$ simply involves fitting the pLSA model to the entire set of training images. In particular it is not necessary to supply the identity of the images (i.e. which category they are in).

Classification of an unseen test image proceeds in two stages. First the document specific mixing coefficients $P(z|d_{test})$ are computed, and following these are used to classify the test images. In more detail, document specific mixing coefficients $P(z|d_{test})$ are computed using the fold-in heuristic described in [64]. The result is that the test image is represented by a Z-vector. The test image is then classified using a K-NN or SVM on the Z-vectors of the training images.

B.3 Datasets and methodology

In order to test our method two public and widely known databases have been used: MIAS -Mammographic Image Analysis Society- database [132] and DDSM -Digital Database of Screening Mammographies- database [62]. Both are explained following:

- **MIAS**. This database is composed by the Medio-Lateral Oblique views of both breasts of 161 women (322 mammographies). The MIAS database provides annotations for each mammogram, and one of them is referred to the breast density. The images are labelled as: (i) *fatty* (106 images) if the breast is almost entirely fatty, (ii) *glandular* (104 images) if the breast contains some fibroglandular tissue, or (iii) *dense* (112 images) if the breast is extremely dense. Moreover, two experts mammographic readers, form the *Hospital Universitari Josep Trueta* of *Girona*, classified the MIAS database according to BI-RADS categories: BI-RADS I (128 images), BI-RADS II (80 images), BI-RADS III (70 images), and BI-RADS IV (44 images). Note that although a strong correlation exists between fatty class and BI-RADS I, glandular and dense tissue are distributed among the rest of BI-RADS categories.

- **DDSM**. We use a set which consists of 500 Medio-Lateral Oblique mammograms from the right breast: BI-RADS I (125 images), BI-RADS II (125 images), BI-RADS III (125 images), and BI-RADS IV (125 images). This database provides for each mammogram additional information, including the density of the breast determined by an expert according to BIRADS categories.

In order to evaluate the results, we used a leave-one-out method, in which each sample is analysed by a classifier which is trained using all other samples. However when working with the MIAS dataset, we leave the two images (left and right breast) from the same woman. This has to be done in order no to bias the results, because both breasts of the same woman have very similar tissue features. Therefore for the MIAS database we use 320 training images and 2 for testing 161 times, changing the test and train images every time. For the DDSM database we use 499 training images and 1 for testing 500 times.

The classification task is to assign each test image to one category. In more detail, when using the K-NN, it selects the K nearest neighbours of the new image within the training database. Then, it assigns to the new mammogram the label of the category which is most represented within the K nearest neighbours. An Euclidean distance function is used. When using the SVM a gaussian kernel is used, and the multi-class classification is done using the one-versus-all rule: a classifier is learned to separate each class form the rest, and a test image is assigned the label of the classifier with the highest response. Overall performance rates are measured by the average value of the diagonal entries of the confusion table.

B.4 Experimental results

We divided this section in three Subsections. The first one shows the results obtained when classifying the MIAS dataset using its own annotation: fatty, glandular and dense. The second one shows the results when BI-RADS annotation is used over both the MIAS and DDSM databases. Last subsection shows a comparison with other works. We investigated the classification performance when using K-NN and SVM classifiers over $P(z|d)$ and when changing the value of different parameters: N (size of the patch when using textons), r (radii of the patch when using SIFT descriptors), M (space between patches), V (number of of visual words of the vocabulary obtained using k-means), K (number of neighbours when using K-NN) and the two descriptors explained in section B.2.2.

B.4.1 MIAS annotation

The best results here have been obtained when $V = 1600$, $Z = 20$ and $K = 6$. Note that K have only sense if K-NN classifier is used. Results increase around 2% when using overlap between patches ($M < N$). Figure B.3 shows the results when classifying using

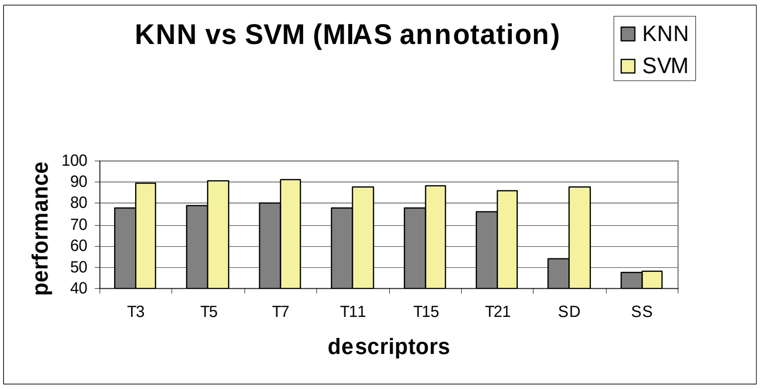


Figure B.3: Performance according to MIAS annotation when changing the values of parameters N and r and fixing $V = 1600$, $Z = 20$ and $M = 2$ for K-NN and SVM, and $K = 6$ for K-NN. $T3$ = Textons with $N = 3$ and so on; SD = SIFT Dense; SS = SIFT Sparse.

the MIAS annotation and the two tested classifiers K-NN and SVM. Results using different descriptors (textons, dense and sparse SIFT) are shown. The best rate classification is obtained when texton vocabulary is used with $N = 7$ and $M = 2$. When using the K-NN the performance is 80.00% and increases up to 91.39% when using SVM. SVM always outperforms the K-NN classifier. The percentages drastically decreases to 54.2% (for K-NN) and to 87.98% (for SVM) when using the vocabulary obtained from the dense SIFT descriptors with $r = 8$ and $M = 5$. This could be due to the nature of this kind of features: they are local histograms of edge directions computed over different parts of the local patch. In all kind of tissues provided from the mammograms there are a lot of edges and changes in the gradient orientation, so in this case, SIFT features are not a good discriminant to classify the tissue density. Better performances have been obtained with dense descriptors and high degree of overlap.

B.4.2 BI-RADS annotation

Best results are obtained when $V = 1600$, $Z = 20$ and $K = 7$. Figure B.4 shows the results when classifying the MIAS and DDSM datasets using BI-RADS annotation, SVM and different descriptors. This annotation is the one that specialists use when classifying the tissue density. For MIAS dataset, best result (95.42%) is obtained when using textons with $N = 7$ and with overlap ($M = 2$). More accurate results are obtained when using

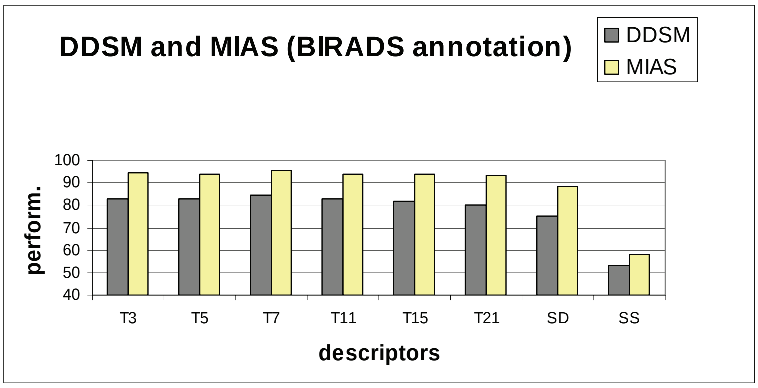


Figure B.4: Performance according to BI-RADS annotation over MIAS and DDSM datasets when changing the values of parameters N, r, M and fixing $V = 1600$, $Z = 20$. SVM are used. $T3$ = Textons when $N = 3$ and so on; SD = SIFT Dense; SS = SIFT Sparse.

	B-I	B-II	B-III	B-IV
B-I	96.06%	3.93%	0%	0%
B-II	5.12%	93.58%	1.28%	0%
B-III	0%	2.85%	94.28%	2.85%
B-IV	0%	0%	2.27%	97.72%

Table B.1: Confusion table when using BI-RADS annotation. Texton vocabulary and $N = 7$, $M = 2$, $V = 1600$, $Z = 20$ are used.

dense SIFT descriptors (88.19%) than when using the sparse ones (58.34%). Best results with DDSM dataset is 84.75% also with $N = 7$ and $M = 2$. Results when using K-NN are around 18% worse.

As can be seen from the confusion matrix of table B.1, the best classified tissue belongs to BI-RADS IV and the most difficult to classify and the ones which present most confusion are BI-RADS II and III. However, following previous works on breast tissue classification according to BI-RADS categories [22, 104, 113], we can reduce this four-class classification problem to the following two-class problem: (BI-RADS I and II) vs (BI-RADS III and IV). In other words, breasts with low density against breast with high density. With this supposition, a classification accuracy of 99.51% and 98.24% respectively is achieved.

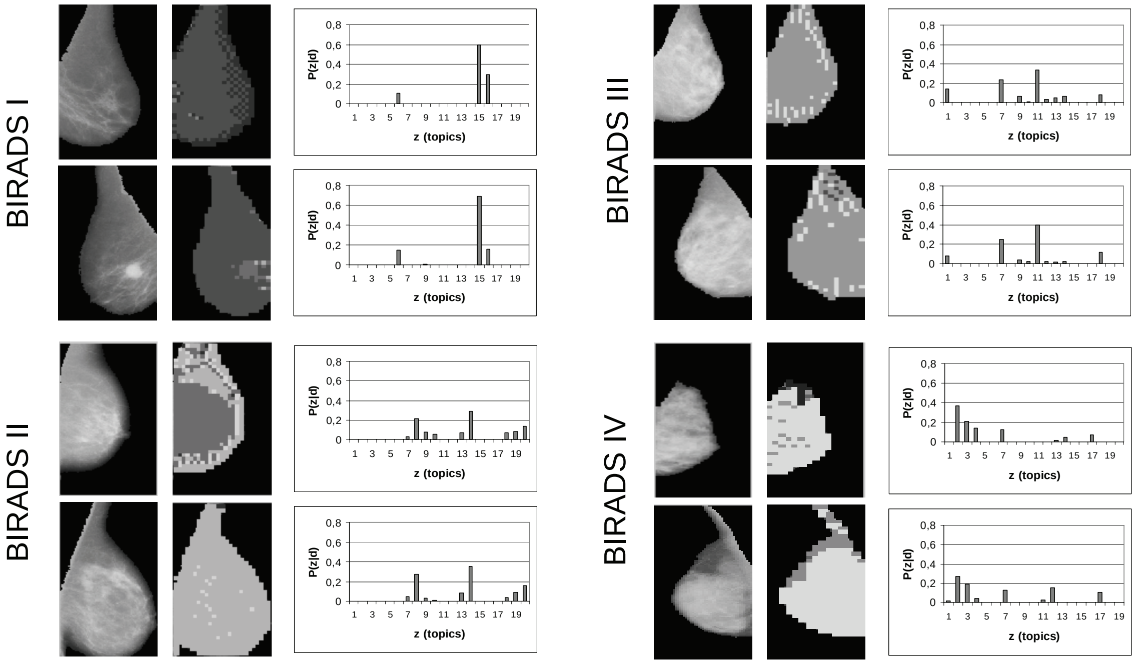


Figure B.5: Segmentation results of two mammograms of each class: BI-RADS I, BI-RADS II, BI-RADS III and BI-RADS IV with the histogram of their topic distribution ($P(z|d)$). The parameters used are: visual textons to compute the vocabulary, $N = 7$, $M = 2$, $V = 1600$ and $Z = 20$.

Figure B.5 shows examples of the spatial distribution of a number of topics (tissue densities) and their histogram of topic distributions ($P(z|d)$). Patches are painted according to the maximum posterior $P(z|w,d)$ (4.6). For each visual word in the image we choose the topic with maximum posterior $P(z|w,d)$ and paint the patch with its associated colour, so each colour represents a different topic (the topic colour is chosen randomly).

The images of figure B.5 are the segmentation of the parenchymal densities in mammograms. They illustrate that topics are representing consistent density tissues across images, there is a similar topic distribution (similar colour) for images from the same BI-RADS category. See for example that images belonging to BI-RADS I are very dark, while images from BI-RADS IV are lighter, showing that there is a different tissue density. If we observe the histograms we can see that those from the images of the same BI-RADS category, have a similar behaviour and topic distribution is consistent across the four BI-RADS categories.

B.4.3 A brief comparison

We can compare the obtained results when using the own annotations of the MIAS dataset with those obtained by Blot and Zwiggelaar [18] and Oliver *et al.* [104]. The first one used a subset of the MIAS database (about 100 images per class) and obtained a 50% of correct classified mammograms. The second one increased this result to 73.00% and used a subset of 60 images per category. Our proposal outperforms both methods obtaining an score of 91.39% of correct classification.

When classifying the MIAS dataset using BI-RADS annotation we can compare the result with [106]. They obtained a 50% of correct classification when classifying the four categories while our performance is of 95.42%. Moreover we improve their results when classifying with only two categories (low and high density). In [106] they obtained 80.00% and we obtain 98.88%.

Bovis and Singh [22] and Oliver *et al.* [105] worked with the DDSM dataset obtaining a 71% and 50% of correct classified images, while working with DDSM we obtain 84.75%. Note that in [105], they only used a subset of 300 images whereas in [22] and our approach used a subset of 500. Other authors classified the tissue density using other datasets. For example Petroudi *et al.* [113] obtained a 76% of correct classified tissues and also works with BI-RADS annotation. However we can not compare this last result directly because their approach was developed by using a different database. Table B.2 summarises these results.

B.4.4 Summary and discussion

We have demonstrated the performance of our approach to classify tissue in mammograms. We investigated performances when working with K-NN and SVM and showed that SVM always outperform the K-NN classifier. We also investigated two kinds of descriptors: textons and SIFT features and our results showed that textons work better over this kind of images. Even though SIFT features have been stated as very useful for object and scene classification, they present a worst performance in our work. This is because SIFT features work with histograms of edge directions and all the tissues in mammograms have a lot of lines. Thus, we can not disambiguate tissue density with this feature (edges). We also have demonstrated that the classification process works better with a high degree

#Ref	Database	Annot.	Author (%)	Our (%)
[18]	MIAS	MIAS	50%	91.39%
[104]	MIAS	MIAS	73%	91.39%
[106]	MIAS	BI-RADS	50%	95.42%
[105]	DDSM	BI-RADS	47%	84.75%
[22]	DDSM	BI-RADS	71%	84.75%
[113]	OXFORD	BI-RADS	76%	–

Table B.2: Comparison summary of the proposed method with other works that classify parenchymal density. Note that the approache of Petroudi *et al.* [113] work with a different dataset and we could not give a direct comparison. MIAS annotation is with 3 classes (fatty, glandular and dense) while BI-RADS annotation is 4 classes (from I to IV).

of overlap between patches.

Best results are obtained with SVM classifier when working with textons vocabulary and $V = 1600$, $Z = 20$, $N = 7$, $M = 2$. Specifically, when classifying with MIAS annotation (3 categories) we obtained a 91.39% of correct classified images. When classifying with the same database with BI-RADS annotation (4 categories) the score obtained is 95.39% and for DDSM dataset the accuracy is 84.75%. We also compared our proposal with several previous approaches that worked with the same databases, and our results outperformed all of them. The main drawback of these techniques is they rely on an initial segmentation of the breast. We think this may be a reason of the superiority of our results. As it is well known, the segmentation is always a very hard task, and specially on medical image. Hence, a wrong segmentation can imply errors on the characterisation and later classification.

B.5 Conclusions

We have demonstrated the successful application of the hybrid system to medical image domain when classifying breast tissue in mammograms. We have represented the images according to their tissue densities and we have shown that the distribution for the same category are similar. Besides, we have studied the influence of various descriptor parameters and have shown that using texture descriptors with overlap works better than SIFT features when working with mammograms.

Bibliography

[1] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, and J. C. Tilton. Learning bayesian classifiers for scene classification with a visual grammar. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):581–589, March 2005.

[2] American Cancer Society. Breast cancer: facts and figures. 2003-04. *ACS*, 2003.

[3] American College of Radiology. *Illustrated Breast Imaging Reporting and Data System BIRADS*. American College of Radiology, 3rd edition, 1998.

[4] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[5] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1300–1305, 1997.

[6] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *International Conference on Machine Learning*, New York, NY, USA, 2004.

[7] K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D.M. Blei, and M.I. Jordan. Modeling words and pictures. *Journal of Machine Learning Research special issue on machine learning methods for text and images*, 3:1107–1135, March 2003.

[8] H.G. Barrow, J.M. Tenenbaum, R.c. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence*, pages 659–663, Cambridge, 1977.

[9] H. and Bay. Surf: Speeded up robust features. In *European Conference on Computer Vision*, volume 1, pages 404–417, Graz, Austria, 2006.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 1988.

[11] A. C. Berg. *Shape Matching and Object Recognition*. PhD thesis, Computer Science Division, University of California., 2005.

[12] A. C. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–33, San Diego, CA, June 2005.

[13] D. Berwick and S.W. Lee. Spectral gradients for color-based object recognition and indexing. *Computer Vision and Image Understanding*, 94:28–43, 2004.

[14] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[15] A. Bissacco, M. H. Yang, and S. Soatto. Detecting humans via their pose. In *Neural Information Processing Systems*, Vancouver, Canada, 2006.

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

[17] C. Blostein and N. Ahuja. A multiscale region detector. *Computer Vision, Graphics and Image Processing*, 45:22–41, 1989.

[18] L. Blot and R. Zwiggelaar. Background texture extraction for the classification of mammographic parenchymal patterns. In *Medical Image Understanding and Analysis*, pages 145–148, Birmingham, UK, 2001.

[19] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988.

[20] A. Bosch, X. Muñoz, and J. Freixenet. Segmentation and description of natural outdoor scenes. *Image and Vision Computing*, 25(5):727–740, May 2006.

[21] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *European Conference on Computer Vision*, volume 4, pages 517–530, Graz, Austria, March 2006.

[22] K. Bovis and S. Singh. Classification of mammographic breast density using a combined classifier paradigm. In *International Workshop on Digital Mammography*, pages 177–180, Bremen, Germany, 2002.

[23] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[24] S.D. Buluswar and B.A. Draper. Color machine vision for autonomous vehicles. *International Journal of Engineering Applications of Artificial Intelligence*, 11(2):245–256, 1998.

[25] S.D. Buluswar and B.A. Draper. Color models for outdoor machine vision. *Computer Vision and Image Understanding*, 85:71–99, 2002.

[26] S. Buseman, J. Mouchawar, N. Calonge, and T. Byers. Mammography screening matters for young women with breast carcinoma. *Cancer*, 97(2):352–358, 2003.

[27] J.W. Byng, N.F. Boyd, E. Fishell, R.A. Jong, and M.J. Yaffe. Automated analysis of mammographic densities. 41:909–923, 1996.

[28] N.W. Campbell, B.T Thomas, and T. Troscianko. A two-stage process for accurate image segmentation. In *IEEE International Conference on Image Processing*, volume 2, pages 655–659, Washington DC, USA, 1997.

[29] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[30] M Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics and Image Processing*, 52:145–170, 1990.

[31] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001.

[32] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description*, 13(1):26–38, 2003.

[33] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, 2007.

[34] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 640–645, San Juan, Puerto Rico, 1997.

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

[36] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, San Diego, California, June 2005.

[37] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[38] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision*, volume IV, pages 97–112, Copenhage, Denmark, 2002.

[39] J. Fan, Y. Gao, H. Luo, and G. Xu. Statistical modeling and conceptualization of natural images. *Pattern Recognition*, 38:865–885, 2005.

[40] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *International Conference on Computer Vision*, volume 2, pages 1134–1141, Nice, France, October 2003.

[41] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, Washington, DC, USA, 2005.

[42] L. Fei-Fei, R. VanRullen, C. Koch, and P. Perona. Rapid natural scene categorization in the near absence of attention. *Proceedings of the National Academy of Sciences of the United States of America*, 99(14):9596–9601, 2002.

[43] P. F. Felzenszwalb. Learning models for object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 56–62, Kauai,USA, December 2001.

[44] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *International Conference on Computer Vision*, volume II, pages 1816–1823, Beijing, China, October 2005.

[45] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, Madison, Wisconsin, 2003.

[46] R. Ferrari and R. Rangayyan. Automatic identification of the pectoral muscle in mammograms. *IEEE Transactions on Medical Image*, 13:232–245, 2004.

[47] G. D. Finlayson, B. Schiele, and J. L. Crowley. Comprehensive colour normalization. In *European Conference on Computer Vision*, volume 1, pages 475–490, Freiburg, Germany, 1998.

[48] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92, January 1973.

[49] C. Fredembach, M. Schröder, and S. Süsstrunk. Eigenregions for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1645–1649, December 2004.

[50] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[51] J. Garding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191, 1996.

[52] D. Gavrila and V. Philomin. Real-time object detection for “smart” vehicles. In *International Conference on Computer Vision*, volume 1, pages 87–93, Kerkyra, Corfu, Greece, 1999.

[53] T. Geodeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. Van Gool. Omnidirectional sparse visual path following with occlusion-robust feature tracking. In *OMNIVIS Workshop, International Conference on Computer Vision*, volume 3115, pages 207–215, Beijing, China, October 2005.

[54] J. Geusebroek, R. van den Boomgaard, and A. Smeulders. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1228–1350, 2001.

[55] R.C Gonzalez and R.E Woods. *Digital Image Processing*. Addison-Wesley, 1993.

[56] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, 3rd edition, Boston, 2001.

[57] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, volume 2, pages 1458–1465, 2005.

[58] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features (version 2). Technical Report CSAIL-TR-2006-020, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, 2006.

[59] G. Griffin, A. Holub, and P. Perona. Caltech 256 object category dataset. Technical Report UCB/CSD-04-1366, California Institute of Technology, 2007.

[60] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[61] D. Haussler. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, 1999.

[62] M. Heath, K. Bowyer, D. Kopans, R. Moore, and P. J. Kegelmeyer. The Digital Database for Screening Mammography. In *International Workshop on Digital Mammography*, pages 212–218, 2000.

[63] A. Hegerath, T. Deselaers, and H. Ney. Patch-based object recognition using discriminatively trained gaussian mixtures. In *British Machine Vision Conference*, volume 2, pages 519–528, June.

[64] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998.

[65] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 41(2):177–196, 2001.

[66] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):267–278, March 1996.

[67] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[68] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[69] N. Karssemeijer. Automated classification of mammographic parenchymal pattern. 28(6):365–378, 1998.

[70] Y. Ke and Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–517, Washington DC, USA, 2004.

[71] J. Koenderink and A. Van Doorm. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[72] S. Kumar, A. C. Loui, and M. Hebert. An observation-constrained generative approach for probabilistic classification of image regions. *Image and Vision Computing*, 21:87–97, 2003.

[73] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 335–344, 1988.

[74] I. Laptev. Improvements of object detection using boosted histograms. In *British Machine Vision Conference*, volume 3, pages 949–958, June.

[75] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 319–324, Madison, Wisconsin, June 2003.

[76] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, June 2006.

[77] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive means-shift search. In *DAGM*, pages 145–153, August 2004.

[78] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[79] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[80] D. Lewis, T. Jebara, and W. Noble. Nonstationary kernel combination. In *International Conference on Machine Learning*, pages 553 – 560, Pittsburgh, PA, 2006.

[81] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision, CVPR*, page 178, Washington D.C., USA, June 2004.

[82] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. pages 389–400, May 1994.

[83] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. *Image and Vision Computing*, 15:415–434, 1997.

[84] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[85] J. Luo, A. E. Savakis, and A. Singhal. A bayesian network-based framework for semantic image understanding. *Pattern Recognition*, 38:919–934, 2005.

[86] J. Luo, A. Singhal, S. P. Etz, and R. T. Gray. A computational approach to determination of main subject regions in photographic images. *Image and Vision Computing*, 22:227–241, 2004.

[87] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:837–842, 1996.

[88] J. Martí, J. Freixenet, J. Batlle, and A. Casals. A new approach to outdoor scene description based on learning and top-down segmentation. *Image and Vision Computing*, 19:1041–1055, January 2001.

[89] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[90] D. Masip, M. Bressan, and J. Vitria. Feature extraction methods for real-time face detection and classification. *Eurasip Journal on Applied Signal Processing*, (13):2061–2071, 2005.

[91] J. Matas, O. Chum, J. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume 1, pages 284–393, 2002.

[92] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, pages 535–531, Vancouver, Canada, 2001.

[93] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[94] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[95] P. Miller and S. Astley. Classification of breast tissue by texture and analysis. *Image and Vision Computing*, 10:227–282, 1992.

[96] A. Mojsilovic, J. Gomes, and B. Rogowitz. Isee: Perceptual features for image library navigation. In *SPIE: Human vision and electronic imaging*, volume 4662, pages 266–277, San Jose, California, June 2002.

[97] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Neural Information Processing Systems*, 2006.

[98] H. Mori, K. Kobayashi, N. Ohtuki, and S. Kotani. Color imipression factor: an image understanding method for outdoor mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 380–387, Grenoble, France, 1993.

[99] H. Movarec. Towards automatic visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence*, pages 584–586, Cambridge, MA, USA, 1977.

[100] J. Mutch and D. Lowe. Multiclass object recognition using sparse, localized features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 11–18, New York, June 2006.

[101] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13:222–241, 1980.

[102] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[103] A. Oliva and A. Torralba. Scene-centered description from spatial envelope properties. In *International Workshop on Biologically Motivated Computer Vision, LNCS*, volume 2525, pages 263–272, Tuebingen, Germany, 2002.

[104] A. Oliver, J. Freixenet, A. Bosch, D. Raba, and R. Zwiggelaar. Automatic classification of breast tissue. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 431–438, Estoril, Portugal, June 2005.

[105] A. Oliver, J. Freixenet, and R. Zwiggelaar. Automatic classification of breast density. In *IEEE International Conference on Image Processing*, volume 2, pages 1258–1261, Genova, 2005.

[106] A. Oliver, J. Martí, J. Freixenet, J. Pont, and R. Zwiggelaar. Automatic classification of breast density according birads categories using a clustering approach. In *Computed Aided Radiology and Surgery*, Berlin, Germany, June 2005.

[107] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.

[108] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *European Conference on Computer Vision*, volume 2, pages 575–588, Graz, Austria, 2006.

[109] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, 2007.

[110] S. Paek and S-F. Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *IEEE International Conference on Multimedia and Expo*, volume II, pages 1133–1136, New York, USA, 2000.

[111] F. Perronin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, volume 4, pages 464–475, Graz, Austria, 2006.

[112] V. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, 2006.

[113] S. Petroudi, T. Kadir, and M. Brady. Automatic classification of mammographic parenchymal patterns: A statistical approach. In *International Conference IEEE Engineering in Medicine and Biology Society*, volume 2, pages 416–423, 2003.

[114] J. Philbin, A. Bosch, O. Chum, J. Geusebroek, J. Sivic, and A. Zisserman. Oxford trecvid 2006 - notebook paper. In *Proceedings of the TRECVID 2006 Workshop*, 2006.

[115] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

[116] P. Quelhas, F. Monay, J.M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, pages 883–890, Beijing, China, October 2005.

[117] J.J. Rocchio. *Relevance feedback in information retrieval.* In the SMART Retrieval System - Experiments in Automatic Document Processing, Prentice Hall, Englewood Cliffs, NJ, 1971.

[118] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(2), 1995.

[119] J. Sanchez, X. Binefa, and J. Vitria. Shot partitioning based recognition of tv commercials. *Multimedia Tools Applications*, 18(3):233–247, 2002.

[120] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *International Conference on Computer Vision*, volume 2, pages 636–643, Bancouver, B.C., Canada, July 2001.

[121] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or how do i organize my holiday snaps? In *European Conference on Computer Vision*, pages 414–431, Copenhagen, 2002.

[122] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE Computer Society Conference on Computer Vision*

and Pattern Recognition, volume 1, pages 746–751, Hilton Head Island, South Carolina, 2000.

[123] N. Serrano, A.E. Savakis, and J. Luo. Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition*, 37:1773–1784, 2004.

[124] J. Shen and S. Castan. An optimal linear operator for step edge detection. *Computer Vision, Graphics and Image Processing*, 54(2):112–133, 1992.

[125] J. Shen, J. Shepherd, and A. H. H. Ngu. Semantic-sensitive classification for large image libraries. In *International Multimedia Modelling Conference*, pages 340–345, Melbourne, Australia, January 2005.

[126] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *International Conference on Computer Vision*, volume 1, pages 503–510, Beijing, China, 2005.

[127] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 235–241, Madison, Wisconsin, June 2003.

[128] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W. T Freeman. Discovering objects and their locations in images. In *International Conference on Computer Vision*, pages 370–377, Beijing, China, October 2005.

[129] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, Nice, France, October 2003.

[130] A.R Smith. Color bamut transform pairs. *Computer and Graphics*, 12(3):12–19, 1978.

[131] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *International Conference on Computer Vision*, volume 2, pages 1063–1070, Nice, France, 2003.

[132] J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, S. Kok, P. Taylor, D. Betal, and J. Savage. The Mammographic Image Analysis Society digital mammogram database. In *International Workshop on Digital Mammography*, pages 211–221, York, England, 1994.

[133] E.B. Sudderth, A. Torralba, W.T. Freeman, and A.S. Willsky. Learning hierarchical models of scenes, objects and parts. In *International Conference on Computer Vision*, volume 2, pages 1331–1338, Beijing, China, 2005.

[134] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *ICCV Workshop on Content-based Access of Image and Video Databases*, pages 42–50, Bombay, India, 1998.

[135] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet process. *Neural Information Processing Systems*, 17:1385–1392, 2005.

[136] J.N. Tenenbaum. An interactive facility for scene analysis research. Technical report, Stanford Research Institute, 1974.

[137] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996.

[138] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *European Conference on Computer Vision*, pages 518–529, Prague, Czech Republic, 2004.

[139] A. Torralba, K.P. Murphy, and W.T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada.

[140] A. Torralba and A. Oliva. Semantic organization of scenes using discriminant structural templates. In *International Conference on Computer Vision*, pages 1253–1258, Korfu, Greece, September 1999.

[141] A. Treisman and G. Gelade. A feature-integration theory of atention. *Cognitive Psychology*, 12:97–136, 1980.

[142] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affinely invariant neighbourhoods. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[143] A. Vailaya, A. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10:117–129, 2001.

[144] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang. Content-based hierarchical classification of vacation images. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 518–523, Florence, Italy, June 1999.

[145] A. Vailaya, A. Jain, and H. Zhang. On image classification: City vs. landscapes. *Pattern Recognition*, 31(12):1921–1935, 1998.

[146] L. Van Gool, T. Moons, and D. Ungureanu. Affne / photometric invariants for planar intensity patterns. In *European Conference on Computer Vision*, pages 642–651, Cambridge, UK, 1996.

[147] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698, Madison, Wisconsin, June 2003.

[148] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, Kauai,USA, December 2001.

[149] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step. In *International Conference on Image and Video Retrieval*, volume 3115, pages 207–215, Dublin, Ireland, July 2004.

[150] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, January 2007.

[151] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1597–1604, New York, June 2006.

[152] J. Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, volume 2, pages 332–348, Graz, Austria, May 2006.

[153] J. Winn and A. Criminisi. Object class recognition at a glance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, USA, 2006.

[154] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 37– 44, New York, USA, 2006.

[155] J.N. Wolfe. Risk for breast cancer development determined by mammographic parenchymal pattern. *Cancer*, 37:2486–2492, 1976.

[156] D. Yagi, K. Abe, and H. Nakatami. Segmentation of color aerial photographs using hsv color models. In *IAPR Workshop on Machine Vision Applications*, pages 367–370, Tokyo, Japan, 1992.

[157] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136, New York, June 2006.

[158] J. Zhang, M. Marszałek, and C. Lazebnik, S. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.

[159] R. Zhang and Z. Zhang. Hidden semantic concept discovery in region based image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 996–1001, Washington, DC, USA, June 2004.

[160] C. Zhou, H.P. Chan, N. Petrick, M.A. Helvie, M.M. Goodsitt, B. Sahiner, and L.M. Hadjiiski. Computerized image analysis: Estimation of breast density on mammograms. *Medical Physics*, 28(6):1056–1069, 2001.

[161] R. Zwiggelaar and E.R.E Denton. Optimal segmentation of mammographic images. In *International Workshop on Digital Mammography*, Chapel Hill, North Carolina, June 2004.