University of London
Imperial College of Science, Technology and Medicine
Department of Computing

Image indexing and retrieval using automated annotation

Alexei Yavlinsky

Submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, June 2007

Abstract

Searching digital information archives on the Internet and elsewhere has become a significant part of our daily lives. Amongst the rapidly growing body of information there are a vast number of digital images. The task of automated image retrieval is complicated by the fact that many images do not have adequate textual descriptions. Retrieval of images through analysis of their visual content is therefore an exciting and a worthwhile research challenge. In this thesis we argue that models of simple image features, such as global colour and texture, can be used to predict instances of different objects and scenes within photographic images. On this basis we propose the use of nonparametric density estimation to model these features and thus endow unlabelled images with probabilities of containing particular objects and scenes. This process, termed "automated image annotation", enables us to set up a scalable image indexing framework that allows users to retrieve unlabelled images from large collections using simple keyword queries. In this thesis we first investigate which image features yield good annotation performance on a number of different test image collections. We pay particular attention to modelling these features effectively. Our experiments show that top benchmark performance results can be rivaled by our approach. Notably, we demonstrate that in addition to enabling retrieval of unlabelled images, our image annotation method can be used for improving the accuracy of text-based Internet image search. We then investigate whether our chosen image features model the presence of objects and scenes in a general and consistent manner. We do so by rigorously comparing the features' characteristic values for similar semantic image categories in different image collections. The investigation results are positive, indicating that our annotation method is sufficiently general. Finally, we show that automatically assigned image annotations can be re-used to improve the accuracy of the initial image annotation index at a small computational cost. This is a useful property for maintaining indexes of very large image collections.

Acknowledgements

During my graduate studies I was extremely fortunate to be surrounded by people who helped me in my research and whom I could ask for advice on how to make progress. I would like to thank:

- my supervisor, Professor Stefan Rüger, for his wise academic advice and encouragement of independent thinking.
- my friends and colleagues in the Multimedia and Information Systems group – Daniel, Peter, João, Simon, Ed, Marcus and Shyamala – for being constant sources of inspiration and invaluable research expertise.
- my friends, Kobus, Charanpal and Uri for their insight and mature advice.
- Universities UK for partially funding my research through the Overseas Research Scholarship.

Finally, and most importantly, I would like to thank my family for their everlasting love, patience and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are a growing number of unlabelled digitised images available on the Internet and in private archives. This growth can be attributed to the ease with which users are able to produce and archive photographs with today's consumer electronic devices. There are good reasons to believe that the rate of this growth will increase in the future, forcing search engine companies such as Google[1] and Flickr[2] to devote vast computational resources to their image search engines. However, reliable textual information *about* these photographs is scarcely available, making the task of building effective image retrieval systems ever more challenging. Photographic archive companies like Getty Images[3] solve this problem by hiring professional librarians to annotate images manually. However, labelling a collection of photographs by hand is a laborious and so a costly process. Furthermore, consistency can be a major problem in manually labelled collections: a human annotator may miss out a relevant description of an image or might disagree with another person over the 'correct' interpretation of that image. Internet search engines extract image descriptions from webpages in which the images are embedded, but this approach is prone to making incorrect associations between an image and its surrounding text.

In this thesis we attempt to investigate the feasibility of automating indexing and retrieval of digital photographs through simple analysis of image content. Earlier efforts in this direction have mostly been focussed on the query-by-image-example paradigm, in which queries are formulated with example images rather than with keywords. In contrast, this thesis builds on two computer science fields – image processing and information retrieval – to propose a framework for providing verbal access to unlabelled photographic collections. We take an applied look at the problem and demonstrate that very simple image properties enable retrieval of such images. This work is distinct from computer vision research that is aimed at explicitly identifying object presence in images. Instead, the objective of this thesis is

[1] http://images.google.com

[2] http://www.flickr.com

[3] http://creative.gettyimages.com

to automatically assign labels to photographs in a manner that bears significant statistical correlation to human judgement.

1.1 Aims and objectives

The objectives of this work are two-fold:

- to investigate whether one can statistically model relationships between photographs and their human annotations using simple image properties
- to investigate whether models of these relationships can be used for retrieving unlabelled photographs from large image collections.

The first objective can be rephrased as to find simple image features that can be used for *automated image annotation* – that is, for generating relevant keywords for photographs automatically. The second objective is concerned with using automatically generated image annotations for indexing and subsequent retrieval of photographs. This thesis is specifically concerned with retrieval of photographic images, and from now on we shall use the terms 'photograph' and 'image' interchangeably.

1.2 Contributions

This thesis makes a number of contributions to the field of content-based image retrieval:

- It is demonstrated that very simple image properties, such as image colour distribution, can be used for achieving state-of-art image annotation results on a standard dataset. This work was originally published in Yavlinsky et al. (2005). Further work in this direction shows that it is possible to exceed state-of-art results using the same approach.
- A novel probability density estimation technique is proposed that is capable of modelling such simple image properties more effectively. This technique is based on the popular Earth Mover's Distance metric. This was also originally published in Yavlinsky et al. (2005)
- Two new, large image datasets are used for rigorous evaluation of image annotation techniques. One is compiled from manually categorised images downloaded from the Internet. The other is a set of images from the Getty Image Archive, together with their original annotations.
- Simple image properties are shown to capture useful and generic image patterns. This is achieved by systematically compairing their characteristic values for similar keywords across substantially different image collections.

- A new image indexing framework is developed. In this framework automatically generated annotations can be re-used for improving retrieval accuracy of predefined concepts, and for enabling retrieval of new concepts that are not included in the annotation vocabulary. This work was originally published in Yavlinsky and Rüger (2007).

1.3 Thesis structure

The rest of this thesis is organised as follows:

Chapter 2 - Background. This chapter describes prior art and some important work done in the research areas of computer vision, information retrieval, content-based image retrieval and automated image annotation. This thesis borrows techniques from these fields and the purpose of this chapter is to allow the reader to place the above contributions in the context of previous work done in these areas.

Chapter 3 - Image Annotation Using Global Features investigates a probabilistic framework for automatically annotating unlabelled images using non-parametric models of distributions of simple image features. A novel probability density estimation technique, based on the Earth Mover's Distance metric (Rubner, 1998), is proposed and evaluated. Two new image collections are described which were specifically compiled to reflect realistic retrieval scenarios. The chapter also demonstrates the use of automated annotation for improving the quality of text-based Internet image search results.

Chapter 4 - Evaluation of Global Feature Reliability. The aim of this chapter is to establish the generality of the image features described in the previous chapter. Additionally, a new measure for predicting the accuracy of modelling different image classes using such features is proposed and evaluated.

Chapter 5 - Efficient Re-indexing of Automatically Annotated Images. This chapter presents a framework for improving the image index obtained by automated image annotation. A technique for fast image re-indexing based on initial automated annotations is proposed and evaluated. The chapter investigates how this technique helps to address the challenges of limited annotation vocabulary size and low annotation accuracies.

Chapter 6 - Conclusions and Future Work discusses the implications of the research results presented in this thesis. The chapter summarises the research contributions contained in this thesis, and, in light of these, suggests areas for future work.

Chapter 2

Background

2.1 Image retrieval

This thesis contributes to the field of content-based image retrieval. This field has been receiving growing attention from the information retrieval and the computer vision communities. It is an appealing idea that traditional text-based information retrieval methods should be soundly combined with computer vision techniques to allow users to search for images that lack associated textual descriptions and tags (collectively described as *metadata*). The immediate problem, however, is that images lack clearly defined semantic units of composition that would be analogous to words in a text document. Without such units the user's query cannot be matched directly against images and needs to be translated into some other representation. Smeulders et al. (2000) give a thorough overview of content-based image retrieval methods aimed at solving this problem. They split the image retrieval task into three different scenarios: category search, target search and search by association. Category search is concerned with idenfying whether images contain a given object. The objective of target search is to locate images that have a particular appearance and the search is typically conducted by using example sketches or images as a query. In association search the user does not have a specific image in mind, and is simply interested in browsing the image database using a mechanism that takes image similarity into account. In this section we review some important work done in each of these categories. In the next section we shall relate this work to automated image annotation.

2.1.1 Category search (object detection)

Identifiable objects within images would be good candidate compositional units for retrieval. However issues such as illumination and pose variation in photographs are formidable obstacles for fully automatic object categorisation. Nonetheless, there has been substantial progress in this direction. There are two distinct approaches to object detection: model-based and statistical. Model-based techniques are based on geometric constraints encoded manally by the detection algorithm designer. These can be supplied in the form of coordinates of characteristic features of an object or in the form of a 'template' that can be matched directly against multiple parts of the image. Statistical detection is concerned with constructing an object classifier that takes image content as input. This classifier is trained on images containing instances of the object.

Early object class recognition research was mainly concerned with face detection. Kanade (1973) was one of the first researchers to consider the problem of localising facial features in mugshot pictures. Candidate feature points, such as the eyes and the nose, are located by finding sharp edges in the mugshot image. These feature points are then confirmed through a rule-based geometric model imposed by the system designer. Govindaraju et al. (1990) proposed a method for face localisation in complex scenes. The shape of the face is modelled manually using several templates. These templates represent simple facial features that can be extracted automatically, and are configured spatially with respect to one another to describe facial geometry. The templates are connected using deformable 'springs' that measure the deviation of the detected points from the ideal configuration; small deviation implies a candidate face in an image. Face detection inspired Vaillant et al. (1994) to propose a more general approach to object localisation in images. Instead of using a manually encoded model of the object, a neural-network classifier is used to judge whether a particular image patch is likely to contain a face. The network is trained on a database of face images and is applied at multiple image scales to detect faces of different sizes. The distinguishing feature of this approach is that in principle it could be applied to types of objects other than faces. The real breakthrough in face detection was achieved by Viola and Jones (2001): they calculate a large number of features in an image window very quickly, and use a greedy search algorithm — known as Boosting (Freund and Schapire, 1997) — to select features that best discriminate faces from non-faces within such windows. Viola and Jones obtained excellent face detection results. However, their approach and those of others has one major limitation: only frontal views of faces could be detected successfully.

Over the past decade the research focus has shifted towards generic object detection by applying statistical learning algorithms to flexible image representations. Papageorgiou et al. (1998) use wavelet coefficients within an image window to decide whether it contains a given object. A Support Vector Machine (SVM) binary classifier is trained on image windows containing the object and on windows that do not. This classifier uses a subset of coefficients that are found to be informative for the given object class. The subset is obtained using a feature selection technique prior to training. Objects are located by applying the classifier within a sliding window on new images. Good results are reported on face and pedestrian detection. Agarwal et al. (2004) automatically construct a vocabulary of distinctive object parts from a set of sample images of the object class of interest. They represent images using spatial configurations of parts from the vocabulary, and train a classifier to detect instances of objects in new images using this representation. The authors applied this approach to detecting side views of cars in urban environments and achieved very good recognition accuracy. They also report reasonable tolerance towards background clutter and mild occlusion. Fergus et al. (2003) model objects as constellation of random parts, located by an interest-region finding algorithm at different scales. A descriptor for each part is obtained by applying Principle Component Analysis (PCA) to the part's vector of pixel values. Shape is represented by the mutual position of the parts. A probabilistic model, estimated on a training set of images, describes the relationship between the parts' appearance, scale and location. This model is used to predict locations of objects in unlabelled images. Ponce et al. (2006) point out that image collections that have been used to evaluate many of these object detection methods are lacking in terms of diversity and scale, which may not reveal the true recognition performance of these methods. Amongst their criticisms are the constrained positions of objects in these datasets and homogeneous background that alone could be used to infer object presence. They also note the difficulty of creating image collections in which many different object types are located within images, which are necessary for training detection algorithms, and suggest ways of overcoming this problem.

2.1.2 Target search (query by example)

Faced with the difficulties of building generic object detectors, most early efforts in content based image retrieval focused on side-stepping this problem altogether within a 'query by example image' paradigm. Here the user's query is no longer a simple statement of desired image content such as 'find images containing an apple', instead example images or sketches of an apple are submitted to a search engine with the aim of retrieving similar-looking images. Images and sketches are typically represented using vectors of low-level colour and texture properties (Swain and Ballard, 1991; Manjunath and Ma, 1996), and similarity between two images is computed as some inverse function of a metric distance between their corresponding vectors.

Jacobs et al. (1995) proposed a mechanism where images similar to a user-submitted colour sketch are retrieved. This approach has been efficiently implemented in Retrievr[1], which allows sketch-based querying of Flickr images. However, this querying mechanism considerably limits the user in expressing his or her information need. A more natural alternative is querying with example images (Faloutsos et al., 1994; Flickner et al., 1995; Smith and Chang, 1996). Unfortunately, this method often fails to capture similarity that could be inferred by humans, a phenomenon that is now commonly referred to as the *semantic gap* (Smeulders et al., 2000). Furthermore, in many cases locating a suitable example for a search may be a difficult task in itself (Roden, 1999).

[1] http://www.retrievr.com

2.1.3 Association search (browsing)

Heesch (2005) provides an excellent overview of image browsing methods in his PhD thesis. In the same thesis he proposes a method of organising images into a pre-computed *network*. The network is constructed in such a way that images that share certain visual properties are linked together. These properties can be combinations of image colour, texture, or layout. By clicking on one of the image's neighbours in the network the user is presented with its respective neighbours. By choosing the neighbour that is most similar to the target image and repeating this process the user can find areas of the network that contain more relevant images. The idea of an image network is much like that of the World Wide Web: similar images should be linked together and one should be able to jump from image to image until one discovers a part of the network with many images of the desired type. Efficiency of this type of browsing also depends on the severity of the semantic gap, as the same kind of low-level visual features are involved.

2.2 Automated image annotation

A number of researchers have recently investigated the use of automated image annotation for querying image databases using text as an alternative to querying with exemplar images. Substantial progress has been made by assuming that users can cope with imperfect retrieval results and performing retrieval of images according to the *probability* of containing a particular concept of interest. This is in contrast to the approaches that explicitly detect object presence, described above. Such probabilities are assigned to unlabelled images based on models of low-level image feature distributions for each concept, and good retrieval performance has been achieved with this approach on a number of different datasets (Duygulu et al., 2002; Jeon et al., 2003; Lavrenko et al., 2003; Feng et al., 2004; Ghoshal et al., 2005). Progress in this direction has thus enabled a number of real-world image and video retrieval systems to take advantage of automated annotation techniques to improve retrieval performance (Lavrenko et al., 2004; Iyengar et al., 2005). In this thesis we take the view that it is a promising idea to treat automatically inferred annotation probabilities as semantic units for information retrieval, analogous to words in text documents. However, as this research field is relatively new, the standard approach for automated image annotation has not yet emerged. The rest of this chapter is structured as follows.

First, we present a number of arguments in defence of using automated image annotation for image indexing and retrieval. We then review related probabilistic and non-probabilistic image annotation techniques separately. We conclude by reiterating the contributions presented in this thesis and differentiate them from these related techniques.

2.2.1 In support of automated image annotation

Automated image annotation vs. expert manual archiving. Enser et al. (2005) highlight two limitations of automated image annotation as compared to traditional manual indexing by experts. The first is that the keywords in the annotation vocabulary have to relate to visible entities within the image, while frequently users submit search requests addressing the *significance* of depicted objects or scenes. An example request would be to 'find a picture of the first public engagement of Prince Charles' (Enser et al., 2005). Clearly only competent archivists could provide adequate metadata to facilitate queries of this type. The authors argue that significance itself is a special case of the general problem of *interpretability* of images, based on their conceptual contents rather than on any visually salient features. The second limitation is the generic nature of keywords in annotation vocabularies; the authors argue that they "appear to have the common property of visual stimuli which require a minimally-interpretive response from the viewer". Enser et al. cite studies showing that search requests for images with features uniquely identified by proper name are very common, where, again, such visual saliency does not play a useful role. The authors opine that, regardless of the advances in image analysis, defining textual annotations will have to be assigned manually.

These limitations are very important and they prompt us to state the goal and the scope of our approach clearly: to enable retrieval of images from vast unlabelled collections, we would like to endow them with indexing units that bear *significant statistical correlation* to human semantic judgements of these images. While we realise that it is not possible to fully automate conceptual interpretation of images, we believe that having 'semantic indexing', to the extent currently possible in the technical sense, is better than not indexing unlabelled images at all, given the prohibitive cost of manual indexing.

Automated image annotation vs. query-by-example search. One could argue that searching with image examples offers greater flexibility than querying using single keywords from a limited annotation vocabulary, or combinations thereof. A query image is not explicitly associated with any concepts and thus – ideally – a similarity-based search would return images sharing as many semantic facets with the query as possible. However, this approach has a number of practical limitations that may make automated annotation a suitable alternative.

- *Computational cost of queries.* Automated image annotation is typically performed offline, so that keyword probabilities are available at query time. This ensures fast response times to verbal queries. Conversely, query-by-example retrieval involves costly similarity computations between query images and a significant number of images in the database. For large image collections such queries can thus be prohibitively time-consuming at runtime.

- *Sample size.* Users may find it difficult to obtain more than a dozen example images for their query. In statistical terms this may often be insufficient to capture the salient properties of the examples. This is because the typical dimensionality of image feature vectors is high compared to the number of example images in the query. On the other hand, a limited annotation vocabulary allows one to gather sufficient numbers of training images to model the keywords, while the offline nature of automated annotation makes modelling keyword features with many training examples practical.

Automated image annotation vs. object detection. Viola and Jones (2001) train an object classifier on sub-images corresponding to segmented objects of a given type. This classifier is then applied by sliding a window across a new image to localise that type of object. There are two possible problems with using this approach for building a semantic image index. First, there is substantially more effort involved in gathering training data. Not only should one identify images containing a given object type, but one must also manually localise that object accurately in each training image. This becomes burdensome when the number of object types is large. Secondly, specific objects often do not play a prominent role individually in identifying certain visual concepts, such as ‘city’, ‘field’ or ‘ocean’. We favour inferring probabilities of concepts by analysing statistical properties of entire images without breaking them down into individual objects, as it is both more scalable and more generic for our purposes. However, it would be interesting to explore the possibility of a systematic synthesis between the two.

2.2.2 Probabilistic image annotation methods

Probabilistic techniques are popular in information retrieval. In text retrieval, it is used as a principled foundation for reasoning under uncertainty about documents matching users’ information needs. There the central idea is to estimate the probability that a document is relevant to the user’s query, and rank documents according to their probabilities (a good overview of this approach can be found in a book by Manning et al. 2007). This is motivated by the theorem that this ranking principle is optimal from the point of view of statistical risk if the probabilities are calculated perfectly (Ripley, 1996). The following probabilistic image annotation methods are likewise supported by this hypothesis.

Co-occurrence Model. Mori et al. (1999) split each image into equal rectangular tiles, extracted joint colour and texture feature vectors for every tile and clustered all such vectors extracted from images in the training collection. Each image tile is labelled with the image’s annotation terms and a token of the tile’s respective cluster. For a word w and each cluster c, a score is calculated as the number of times the word and the tiles from that cluster co-occur in training images. This score is then normalised so that all word scores add to 1 within each cluster. An unseen image is annotated by likewise segmenting it into tiles and finding the nearest cluster for each of the tiles. The word scores from these clusters are then averaged, and the new image is annotated with the top n words. Mathematically, this is equivalent the following set of equations:

$$p(w|c) = \frac{m_{w,c}}{\sum_w m_{w,c}}, \tag{2.1}$$

where $m_{w,c}$ is the number of times the word w cooccurs with a tile from cluster c, and

$$s(w, x) = \frac{1}{|x|} \sum_{t \in x} p(w|c_t), \tag{2.2}$$

where $s(w, x)$ is the relevance score of w for an unseen image x, c_t is the nearest cluster of x's tile t and $|x|$ is the number of tiles in x.

Translation Model. A popular approach in automated image annotation has been to use an image segmentation algorithm to divide images into a number of irregularly shaped 'blob' regions and to operate on the low-level features of these blobs. Duygulu et al. (2002) created a discrete 'vocabulary' of clusters of such blobs across an image collection and applied a model, inspired by machine translation, to translate between the set of blobs comprising an image and annotation estimates translation probabilities $p(a_{nj} = i)$ that in image n, a particular blob b_i is associated with a specific word w_j, that maximise the likelihood on the training set. The likelihood function is defined as

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) t(w = w_{nj} | b = b_{ni}), \tag{2.3}$$

where N is the number of images and M_n and L_n are the number of words and blobs associated with image n, respectively. The authors maximise this likelihood by learning translation probabilities with the Expectation-Maximisation (EM) algorithm. Once the above translation table is estimated, an unseen image is annotated by picking the most likely word for each of its blobs. The model was evaluated on a dataset consisting of 5,000 images from Corel Stock Photo library. Each image was also assigned 1–5 keywords from a vocabulary of 371 words. The blob features are:

- area
- convexity
- moment of inertia
- average colour
- colour variance
- texture orientation

Cross Media Relevance Model. Jeon et al. (2003) reworked image annotation into a problem in cross-lingual information retrieval, applying a Cross-Media Relevance Model (CMRM) to perform image annotation and ranked retrieval. Similarly to the Translation model (Duygulu et al., 2002), CMRM treats words and clutered blobs as two different lexicons, however instead of seeking a probabilistic translation table it simply models the probability of observing a blob b_j together with a word w_i in a given image. Provided an unlabelled image x is composed of m blobs $x = \{b_1, \ldots, b_m\}$, the probability of a word w_i is calculated as

$$p(w_i|b_1, \ldots, b_m) = \frac{p(w_i, b_1, \ldots, b_m)}{p(b_1, \ldots, b_m)}. \tag{2.4}$$

The joint probability of the word and the blobs can be calculated by summing their joint probability under the *relevance model* of each training image across the entire training set:

$$p(w_i, b_1, \ldots, b_m) = \sum_{y \in T} p(y)p(w_i|y) \prod_{j=1}^{m} p(b_j|y), \tag{2.5}$$

where y is an image in the training set T. This assumes that given an image y words and blobs are generated independently. The probabilities of a word w_i and a blob b_j appearing in the training image y are calculated according to the multinomial probability distribution, respectively, as

$$p(w_i|y) = \alpha \frac{n_{w_i,y}}{N_{w,y} + N_{b,y}} + (1 - \alpha)p(w_i) \tag{2.6}$$

and

$$p(b_j|y) = \beta \frac{n_{b_j,y}}{N_{w,y} + N_{b,y}} + (1 - \beta)p(b_j). \tag{2.7}$$

$n_{w_i,y}$ and $n_{b_j,y}$ are the numbers of times w_i and b_j appear in y and $N_{w,y} + N_{b,y}$ is the total number of words and blobs in the image. $p(w_i)$ and $p(b_j)$ are the background probabilities of w_i and b_j in the training set, and α and β are the smoothing factors. Jeon *et al.* describe these two probabilities as the "relevance model" of image y – a unique model that generated that image. This model is equivalent to imagining y as a bag from which blobs and words can be drawn with replacement and in any order.

To annotate a new image x one can assign to it its n most probable words under this model. Alternatively, one can sample n times from the word distribution of x. The authors report a substantial improvement in annotation accuracy of CMRM over the Translation model. Furthermore, one can perform text-based ranked retrieval. For an m-word query $Q = \{q_1, q_2, \ldots, q_m\}$ the retrieval score for an image x is defined as

$$p(q_1, q_2, \ldots, q_m|x) = \prod_{i=1}^{m} p(q_i|x). \tag{2.8}$$

Good mean average precision results are reported on the dataset used in Duygulu et al. (2002), using the same blob features.

Continuous Relevance Model. One of the disadvantages of the Translation model and CMRM is the need to cluster blobs, which is bound to result in the loss of useful information in image regions. Continuous Relevance Model (CRM) by Lavrenko et al. (2003) is an extension of CMRM in which $p(b|y)$ is expressed as a product of continuous probability density functions. Each blob b_j has now an associated d-dimensional feature vector v_j. In CRM, the joint probability of a set of words w and a set of blobs b is defined as

$$p(w, b) = \sum_{y \in T} p(y) \prod_{i=1}^{|w|} p(w_i|y) \prod_{j=1}^{|b|} p(b_j|y). \tag{2.9}$$

The probability of a blob b_j according to y's relevance model is

$$p(b_j|y) = \int_{\mathbb{R}^d} p(b_j|v)p(v|y)dv, \tag{2.10}$$

where $p(v|y)$ is the probability density function for a feature vector v in image y and $p(b_j|v)$ is an indicator function, which is set to 1 when $v = v_j$ and to zero otherwize. In this case the above equation simplifies to

$$p(b_j|y) = p(v_j|y). \tag{2.11}$$

$p(v|y)$ is a nonparametric kernel-based density estimate:

$$p(v|y) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-(v-v_i)^\intercal \Sigma^{-1} (v-v_i)}, \tag{2.12}$$

where n is the number of blob vectors in image y and $|\Sigma|$ is the covariance matrix used to control the degree of smoothing. Simply put, the probability of a blob b_j occuring in y in CMRM has been replaced with a smoothed probability density estimate of its corresponding vector v_j. Feng et al. (2004) put forward a further modification of the above model where $p(w_i|y)$ is estimated using a Bernoulli distribution. They termed this model Multiple Bernoulli Relevance Model (MBRM). The authors report that both CRM and MBRM significantly outperform CMRM on the standard dataset of Duygulu et al. (2002).

Latent Variable Models. Blei and Jordan (2003) proposed a number of different latent variable models that assume a mixture of latent factors are used to generate words and blobs, the latter represented by real valued vectors of their colour and texture properties. The simplest model of this class is the *Gaussian-multinomial mixture model*, in which a single discrete latent variable is used to represent a joint clustering of an image and its captions. The generative process of this model can be described as follows:

- Pick a latent variable z according to $p(z|\lambda)$
- Repeatedly sample N blob vectors r_n according to $p(b_n|\mu, \Sigma, z)$
- Repeatedly sample M words w_m from the multinomal distribution $p(w_m\beta, z)$

In other words, a latent variable represents parameter settings for one multinomial and one multivariate Gaussian distributions. The parameters of this model with k latent variables can be learned on the data using EM. The joint distribution of a set of words $w = \{w_1, \ldots, w_n\}$ and a set of blobs $b = \{b_1, \ldots, b_n\}$ with the hidden variable z is given as

$$p(w, b, z) = p(z|\lambda) \prod_{n=1}^{N} p(b_n|\mu, \Sigma, z) \prod_{m=1}^{M} p(w_m, |\beta, z). \tag{2.13}$$

Furthermore, the conditional probability of a word given a set of blobs can also be computed by using the Bayes rule to find $p(z|b)$ and summing over the latent variables:

$$p(w_i|b) = \sum_{z} p(w_i|z)p(z|b). \tag{2.14}$$

This model assumes that words and blobs are generated independently given a latent variable, therefore it cannot be used for establishing word-blob correspondence. However, in the same paper the authors provide an extension to the above model where blobs can be labelled with keyword probabilities individually.

Global Scene Modelling. The annotation schemes above use image segmentation algorithms to partition images into their constituent pseudo-objects; statistical models of their co-occurrence with annotation words are then found. The success of models based on this approach would depend, to a large degree, on the accuracy of the image segmentation algorithms. Sidestepping this problem altogether, Torralba and Oliva (2003) use global features for classifying natural images into semantic categories and for predicting the presence of objects. Although a counterintuitive suggestion at first, their approach is underpinned by the following findings of earlier psychovisual studies, which they cite:

- A strong relationship exists between the scene category of a picture and the distribution of coloured regions in the image.
- The background scene around an object is typically constrained and exhibits the texture and colour pattern that is common to environments where the object is usually found.
- The statistics of natural image categories depend not only on the physical characteristics of the world, but also on the viewpoints that observers adopt; each scene type constrains the viewpoints that can be taken by the observer.

Inspired by these findings, they build reliable scene classifiers and use scene information to infer whether an image contains a given object. They note that different scene types have characteristic outputs in the frequency domain and apply Principal Component Analysis and Linear Discriminant Analysis to these outputs to accurately discriminate between man-made and natural scenes. They also show that using features consisting of as few as 16 first principal components of the image frequency signal it is possible to predict with reasonable accuracy the presence of object types as specific as ‘animals’, ‘people’ and ‘vehicles’. They achieve this by applying mixtures of Gaussians to these features to learn the probability density function of each object class.

Other relevant probabilistic image annotation work. Wang and Li (2002) train a 2-D multi-resolution Hidden Markov Model for each keyword to capture low-level feature dependencies across adjacent blocks at progressively higher resolutions in labelled images, subsequently using these models to assign keyword probabilities to unlabelled images. Ghoshal et al. (2005) annotate images using a Hidden Markov Model in which states represent keywords conditioned upon low level image features found in rectangular image blocks. Carneiro and Vasconcelos (2005) likewise split each image into several blocks, and model each word class as a hierarchical mixture of Gaussians describing the JPEG Discrete Cosine Transform (DCT) coefficient information of these blocks. At the time of writing their approach yields the best published results on the dataset of Duygulu et al. (2002); however it is unclear whether this is due to their probabilistic modelling approach or the DCT features they use. Jin et al. (2004) note that in many automated annotation approaches words are predicted independently of one another, and relationships of similar words such as ’grass’ and ’vegetation’ are not taken into account. They propose a language modeling approach that takes such relationships into account when inferring probabilities of keywords based on image content. Jeon and Manmatha (2004) use Maximum Entropy to model relationships between quantized image patch features and image annotations. This is done as an alternative to the translation model of Duygulu et al. (2002), and the authors report significant improvement over the latter approach. Another improvement to the translation model is reported by Kang et al. (2004). The authors propose a modification that increases precision of words that occur infrequently in the training data. Fan et al. (2004) formulate automated image annotation as a two-stage process. During the first stage image blobs are classified into low-level natural scene categories such as ‘sky’, ’grass’ or ‘rock’. An SVM classifier is trained for each category on manually labelled blobs. The second stage consists of estimating a model that associates SVM region category *predictions* to higher level concepts such as ‘garden’ and ’beach’. This is done using a probabilistic model. The central hypothesis of their method is that higher level semantic categories should be modelled in terms of lower level scene-oriented concepts, instead of modelling the former directly in visual feature space. However, it is unclear whether this extra modelling step is inherently advantageous. Metzler and Manmatha (2004) segment training images, connecting them and their annotations in an inference network, whereby an unseen image is annotated by instantiating the network with its regions and propagating belief through the network to nodes representing the words. Magalhães and Rüger (2007) create a high-dimensional image feature space by clustering low-dimensional colour and texture image features into a large number of clusters. The probability density of these low-dimensional features under each cluster’s model becomes a feature component in the high-dimensional space. The authors select the optimal number of such components using the Minimum Description Length criterion and learn the keyword models in this feature space using Logistic Regression.

2.2.3 Non-probabilistic image annotation methods

Methods inspired by the Vector Space Model. In text retrieval, the Vector Space Model (VSM) framework is a popular alternative to probabilistic retrieval (Salton et al., 1975). In this framework, text documents are represented as vectors. Every component of this vector corresponds to the occurrence frequency of a particular word (term) in the document. The user's query is represented in the same manner. Given a query, the documents are ranked according to the similarity of their vectors to the query vector. A popular similarity measure the cosine function of the two vectors. It is also possible to find similar terms within a document collection. If one were to concatenate the document vectors into a matrix column-wise, rows of this matrix can be treated as *term vectors*. Pairwise term similarity can be established by applying the cosine function to the term vectors. Tang and Lewis (2007) adapt this technique for identifying relevant annotations for individual image blobs. This goal is similar to that of the Translation Model Duygulu et al. (2002), which, in addition to predicting relevant annotations, associates them with individual image regions. In their framework, each image in the training set is a dimension (much like each document is a dimension of the term vectors). Both image blobs and annotation keywords can be mapped into this space. A keyword w is mapped onto a vector v_w, in which the component $v_w^{(j)}$ is set to 1 if the training image $I^{(j)}$ is labelled with that keyword; otherwise it is set to zero. A region r is mapped onto a vector v_r. $v_r^{(j)}$ is set to the cosine similarity between the low-level feature vector of r and that of its closest region in image $I^{(j)}$. Now both v_w and v_r are essentially term vectors. Given a new region r', most relevant keywords can be found by computing cosine similarity between $v_{r'}$ and all keyword vectors and choosing n closest keywords. Images are first partitioned by a standard image segmentation algorithm. Each segment is then described by a histogram of quantised Scale Invariant Feature Transform (SIFT) descriptors. These are highly selective feature points that characterise sharp changes in image intensity, such as corners or textured patches. A more detailed discussion of SIFT is presented in Chapter 3.

Words that frequently appear together in text documents are likely to refer to similar concepts. It is possible to use term co-occurence to improve text retrieval. In the VSM framework this is achieved by modifying the similarity function so that increased similarity is assigned to documents containing words that are related to the query terms but do not match them exactly. Latent Semantic Indexing (LSI) is a linear-algebraic approach that establishes such co-occurrences and incorporates them into the similarity measure between documents simultaneously (Deerwester et al., 1990). Singular Value Decomposition (SVD) is performed on the term-by-document matrix and the results of this decomposition are used to project document vectors into a lower-dimensional space. The nature of this projection is such that dimensions corresponding to related terms are projected onto same dimensions in the lower-dimensional representation. It thus becomes possible to retrieve documents that contain related terms to the keywords in the query. When a translated corpus is available for the document collection, LSI can be used for cross-language information retrieval (Landauer and Littman, 1990). This is simply achieved by applying SVD to the term-by-document matrix in which term vectors contain terms from both languages. This results in a lower-dimensional document vector space, in which word co-occurrences across languages are taken into account. It is then possible to map documents that are only available in one of the languages into this space and query them using keywords from the other language. Hare et al. (2006) extend this approach to keyword-based image retrieval of unlabelled images. They treat each image as two parallel documents: one in the language of human annotations, and the other in a 'visual language' of low-level image features. They experiment with two types of features for creating the visual language on two different datasets: quantised SIFT descriptors on the Washington dataset (University of Washington, 2004) and a simple RGB histogram on the Corel dataset used by Duygulu et al. (2002).

Classification approaches. Chapelle et al. (1999) were one of the earliest to classify images from the Corel collection using a binary classifier. The images were represented using different colour histograms and SVMs were used to classify these images into several categories. At this point it is worth noting that in an information retrieval setting one would expect comparatively few relevant images for each class, which is not the case in this paper. Ng et al. (2007) use SVMs and Neural Networks to classify image content in the Corel dataset using MPEG-7 feature descriptors (Manjunath, 2002). Mrowka et al. (2004) likewise classify a mixture of images from Corel and the Internet into several categories using a Radial Basis Function Neural network. They propose a statistical feature selection method that effectively reduces dimensionality of the MPEG-7 visual feature space while maintaining a high classification accuracy. Gao et al. (2006) propose a method for image region classification based on multiple SVM classifiers. Each classifier is trained to discriminate the presence of a particular concept within a homogenous subset of visual features. Outputs of these classifiers are then combined using the Boosting algorithm (Freund and Schapire, 1997). Its objective to combine these classifier outputs in such a way so as to maximally improve over the best single classifier.

2.2.4 Retrieval accuracy evaluation

As mentioned in Section 2.2, automated image annotation benefits from the assumption that users are satisfied when images are ranked by the probability of depicting a desired concept. Naturally we would like to measure the quality of such rankings to compare different image annotation methods. A number of standard information retrieval evaluation measures are at our disposal for this purpose. Given a ranked set of items in response to a query:

$$\text{Recall} = \frac{\text{number of relevant items retrieved}}{\text{total number of relevant items}} \tag{2.15}$$

and

$$\text{Precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}} \tag{2.16}$$

Precision and Recall are set based measures that are useful for evaluating the accuracy of the top n results, but they do not indicate whether most of the relevant items appear near the top of the list. Mean average precision is a single measure that takes the latter aspect into account. It is defined as:

$$\text{Average precision} = \frac{\sum_{r=1}^{N} \text{Precision}(r)}{N}, \tag{2.17}$$

where Precision(r) is the precision of the part of the ranked list that has the r^{th} of the N relevant items as its last item. This measure has been extensively used by the Text REtrieval Conference (TREC) community and we shall use it throughout the thesis.

2.2.5 Thesis contributions revisited

In this chapter we have reviewed important work done in image retrieval and automated image annotation. We have separated the latter into probabilistic and non-probabilistic methods. Probabilistic methods model visual feature distributions of keywords explicitly. In principle, these methods have the potential to predict relevant image keywords more accurately. However, complexity of methods that estimate probability densities of features in a nonparametric manner (e.g. Jeon et al. 2003; Lavrenko et al. 2003; Feng et al. 2004), scales linearly in the number of training examples for each keyword. This results in high computational cost at annotation time. Classification techniques and methods inspired by the Vector Space Model typically seek linear combinations of visual feature components that discriminate best between different image classes (Hare et al., 2006; Ng et al., 2007; Magalhães and Rüger, 2007). The resulting annotation models incur a much smaller computational cost at annotation time but are potentially inferior to the nonparametric probabilistic methods in terms of accuracy. There have been attempts to reduce the computational complexity of the latter through the use of Hidden Markov Models (Wang and Li, 2002; Ghoshal et al., 2005) and Gaussian Mixture Models (Carneiro and Vasconcelos, 2005); we also consider the complexity-reduction problem in this thesis.

However, whilst many of the approaches reviewed in this chapter focus on different ways to model image-keyword relationships (Duygulu et al., 2002; Jeon et al., 2003; Lavrenko et al., 2003; Feng et al., 2004; Ghoshal et al., 2005; Blei and Jordan, 2003; Wang and Li, 2002; Carneiro and Vasconcelos, 2005; Jin et al., 2004; Jeon and Manmatha, 2004; Kang et al., 2004; Fan et al., 2004; Metzler and Manmatha, 2004; Tang and Lewis, 2007; Hare et al., 2006; Gao et al., 2006), we take a step back and investigate the feasibility of automating image annotation using very simple colour and texture image properties. Using these features, we sidestep the issues of image segmentation and labelling individual image parts with keywords. We report that state of art results can be rivaled with such features. We use a very simple probabilistic framework for image annotation. Yet within this framework we propose a novel probability density estimation technique, based on the Earth Mover's Distance metric that takes better advantage of these features compared to traditional statistical methods.

Most results are reported on test collections which are very similar to the training set used to estimate image-keyword relationships, e.g. the Corel dataset. We construct two additional, large datasets to support more realistic evaluation of accuracy. We investigate whether the simple image features extract generic patterns that are useful for all of our datasets. We develop a method for predicting whether a set of such features is adequate for modelling a particular concept. We explicitly explore how annotation accuracy is affected when the collection of images used for training comes from a different source to images that need to be indexed: for example using Corel image collection to estimate keyword models that are then used to annotate images downloaded from the Internet. Noting that such image collection discrepancies can result in the loss of annotation accuracy, we propose a computationally efficient way of ameliorating this effect. This is done by applying well-known text retrieval techniques to image collections which have already been automatically annotated. To summarise, this thesis is concerned with three novel issues:

- *simplifying* the task of automated image annotation at the low-level feature level by utilising appropriate statistical models
- ascertaining the generality of simple image features by investigating annotation accuracy across a number of large, realistic image collections
- developing a framework for computationally efficient improvement of annotation accuracy in collections that have *already* been annotated.

Chapter 3

Image Annotation Using Global Features

3.1 A simple probabilistic annotation model

Suppose a human annotator is prompted for a single annotation word for the image x, and that he chooses word w with probability $p(w|x)$. We wish to model this process. We use Bayes' Theorem to invert the conditional dependence as:

$$p(w|x) = \frac{f(x|w)p(w)}{f(x)}, \tag{3.1}$$

where we interpret $f(x)$ as the probability density of image x and $f(x|w)$ as density of x conditional upon the assignment of annotation w. We now wish to model $f(x|w)$ for each possible annotation word w by collecting a sample T_w of images with each label w as a training set. A critical factor in modeling the densities $f(x|w)$ will be choosing a representation x for the images. In general, we want a representation for which the densities are as separable as possible for different annotation classes w, yet are dense enough for reliable inference from a small sample of images for each class. We consider two different representations: as a vector of real-valued image features $x = (x_1, \ldots, x_d)$, $x_i \in \mathbb{R}$; and as a *signature* of image features, defined later in this section. In the case where x is represented by a real-valued vector, results in statistical pattern recognition tell us that to observe the above conditions we must seek a representation with the lowest possible number of dimensions (Bishop, 1996). This guiding principle, complemented by our desire for a practical image annotation system, motivated us to employ very simple image features.

One method of inference is to specify a parametric form *a priori* for the true distributions of image features for the annotation class w and then estimate the parameters using the methods of classical statistics. Another method is to encode all our knowledge about the true distribution as constraints on the model and choose the model subject to these constraints with maximum entropy (the 'flattest') or minimum relative entropy to some prior density. A third method is to adopt a nonparametric estimator of the true density that makes no prior assumptions about the true density.

The first method is less appropriate within this framework than the second two. In general, the distributions of image features will have shapes that are irregular, not resembling any simple parametric form. Instead we hope this irregularity will be helpful in characterizing and distinguishing the distributions under different word classes. Maximum Entropy is an attractive framework for modelling probability density functions in a principled manner, however parameter estimation for this type of models can be computationally challenging (Schofield, 2006). In this chapter we consider the third method, nonparametric density estimation.

A number of previous approaches have reflected our intuition behind this choice. Consider, for example, Equations 2.9 and 2.14, of the Continuous Relevance Model and of the latent variable model, respectively. The former defines a separate probabilistic "relevance" model for each individual image that generates the visual part of the image (blob feature vectors) and its keywords. The probability distribution over keywords in the vocabulary for an unlabelled image is computed under the relevance model of every training image separately, and the average probability for each keyword is then taken. This approach is nonparametric at heart because the entire training set is used to infer keyword probabilities for the unlabelled image. The latent variable model is conceptually very similar with the exception that a fixed number of models are responsible for generating image blob vectors and keywords, for which the EM algorithm is used to estimate parameters. Our way of formulating the annotation problem, originally published in Yavlinsky et al. (2005), is most closely related to that of Carneiro and Vasconcelos (2005). They also view it as a probabilistic process described by Equation 3.1, in which each word class is independently modelled using a probability density function in low-level feature space. However, instead of using nonparametric density estimation they employ a hierarchical mixture of Gaussians.

3.1.1 Nonparametric density estimation techniques

The simplest nonparametric estimator of a distribution function is the empirical distribution function, but it is known that smoothing can improve efficiency for finite samples (Reiss, 1981). 'Kernel smoothing', first used by Parzen (1962), is a general formulation of this. Where x is a vector $(x_1, \ldots, x_d)$ of real-valued image features, we define the kernel estimate of $f_w(x) = f(x|w)$ as

$$\hat{f_w}(x, h) = \frac{1}{nC} \sum_{i=1}^{n} k(x, {x_w}^{(i)}; h), \tag{3.2}$$

where $x_w^{(1)}, \ldots, x_w^{(n)}$ is the sample of images with label w in the training set T_w, where k is a kernel function that we place over each point $x^{(i)}$, and where $C = \int k(x, \cdot; h)dx$ so that $\hat{f}(x, h)$ integrates to one and is itself a probability density. We omit the subscripts w for the rest of this section to simplify the notation. Here the positive scalar h, called the bandwidth, reflects how wide a kernel is placed over each data point. Under some mild conditions, $\hat{f}$ converges to f in probability as $n \to \infty$ (Härdle, 1992).

Kernel smoothing in real vector space

For real-valued image feature vectors considered the commonly used d-dimensional Gaussian kernel

$$k_G(x, x^{(i)}; h) = \prod_{l=1}^{d} \frac{1}{\sqrt{2\pi h_l}} e^{-\frac{1}{2}\left(\frac{x_l - x_l{}^{(i)}}{h_l}\right)^2}, \tag{3.3}$$

and the d-dimensional Laplace kernel

$$k_L(x, x^{(i)}; h) = \prod_{l=1}^{d} \frac{1}{h_l} e^{-\frac{|x_l - x_l{}^{(i)}|}{h_l}}, \tag{3.4}$$

where $x_l - x_l{}^{(i)}$, the difference between l^{th} components of the feature vector x and the basis point $x^{(i)}$. We set each bandwidth parameter h_l by scaling the sample standard deviation of feature component l by the same constant λ.

Kernel smoothing in Earth Mover's Distance metric space

Multidimensional distributions are often used in image retrieval to describe and summarize different features of an image. A common way to represent such distributions is by quantising the multidimensional feature space into bins and turning image histograms into real valued feature vectors. A fine, regular quantisation of the underlying feature space may result in feature vectors of high dimensionality (e.g., quantising three image colour channels into 8 bins each will result in 512 vector components). Friedman et al. (1984) point out that kernel smoothing may become unreliable in high-dimensional spaces due to the problem known as the *curse of dimensionality*. Therefore, for higher-dimensional feature spaces sufficiently fine quantisation may become incompatible with effective density estimation.

Friedman et al. (1984) examine a projection pursuit method for reducing the effective dimensionality by projecting a space onto a single dimension in a way that preserves its most salient characteristics. This is one way of sidestepping the problem, but we consider another way based on comparing image *signatures* under the Earth Mover's Distance (EMD) measure (Rubner, 1998), which has found several applications in image retrieval (Rubner et al., 2001).

A signature is essentially a cluster-based representation of a multidimensional distribution, defined as $s = \{(c_1, m_1), \ldots, (c_d, m_d)\}$, where, for a cluster i, c_i is the cluster's centroid and m_i is the number of points belonging to that cluster or its mass. Given two such signatures $p = \{(p_1, x_1), \ldots, (p_m, x_m)\}$ and $q = \{(q_1, y_1), \ldots, (q_n, y_n)\}$, EMD is defined as the minimum amount of work necessary to transform one signature into the other. More specifically, we define the cost matrix $D = [d_{ij}]$, where d_{ij} is the distance between p_i and q_j under some fixed metric (termed the *ground distance*). We then seek a flow matrix $F = [f_{ij}]$, where f_{ij} is the flow of mass between p_i and q_j, that minimises the overall cost (work) function $\sum_{i,j} f_{ij} d_{ij}$, assuming the maximum possible mass has to be transferred from p to q. The solution can be obtained using the Transportation Problem optimisatoin algorithm (Rubner, 1998).

One can view a signature as an irregularly quantised histogram, in which the Voronoi partitions of the feature space induced by the signature's centroids define the histogram's bins. Rubner (1998) reports that EMD is more robust than traditional histogram comparison techniques for computing image similarity. In particular they show that images represented with as few as 8 clusters of CIE*Lab* colour outperforms the traditional distance measures applied to high-dimensional colour histograms. They attribute this to the fact that EMD incorporates nearness between histogram bins in the feature space and is thus much less sensitive to choice of quantisation scheme. It has been shown that EMD is a true metric when the total mass of each signature is equal to one (Rubner, 1998; Levina and Bickel, 2001).

Although it has been shown that EMD is a true metric, few properties are currently known about the spaces that can induce. In real vector spaces concepts such as the mean of two vectors and integration are well defined. In the EMD space, however, points are represented by signatures, for which these concepts are not yet defined by mathematicians. Therefore, for reasons of practicality, we are forced to solely rely on EMD values for operations such as clustering.

As EMD captures similarity of multidimensional feature distributions more faithfully than traditional techniques that treat histograms as real-valued vectors, we would like to estimate densities of word classes in the space induced by the EMD metric. The simplest way to perform smoothing in this space is by defining a kernel that asymptotically decreases with the EMD from the signature on which the kernel is centered. We define the *EMD kernel* as

$$k_E(s, s^{(i)}; h) = \frac{1}{h} e^{-\frac{d(s,s^{(i)})}{h}}, \tag{3.5}$$

where $d(s, s^{(i)})$ is the value of EMD between signatures s and $s^{(i)}$, and h is the kernel bandwidth. This kernel places a density function around each basis signature $s^{(i)}$. For $\hat{f}(x)$ to be a proper density function

$$\int_s e^{-\frac{d(s,s^{(i)})}{h}} \tag{3.6}$$

has to equal some constant C regardless of the basis signature $s^{(i)}$ so that

$$\int_s k_E(s, s^{(i)}; h) ds \tag{3.7}$$

can be normalised to equal one. It is nontrivial to ascertain whether the function (3.6) indeed integrates to a constant because $d(s, s^{(i)})$ is a result of a numerical optimisation procedure for each s. This prevents simple analytical integration of the function over all possible signatures. Our approach *assumes* that if the integral results do vary depending on the basis signature, this variance is small enough for reliable probabilistic classification according to Equation 3.1. While we do not investigate this issue further here, we acknowledge that in the long term it is important to study the properties of this integral to establish the theoretical soundness of density estimation in EMD metric space.

Kernel bandwidth optimisation

Several methods have been studied for choosing the optimal bandwidth h for a given kernel and density estimation task. Jones et al. (1996) and Loader (1999) give a good overview. We use the simple method of cross-validation, choosing the bandwidth that maximizes performance on a withheld data set. The exact bandwidth value is obtained by performing golden search optimisation (Press et al., 1986).

Data reduction for density estimation in EMD metric space

The complexity of inference via kernel smoothing grows linearly with the size of the training set, making it a computationally costly choice for image annotation. This has inspired researchers to use mixtures of Gaussians (Carneiro and Vasconcelos, 2005) or Hidden Markov Models (Ghoshal et al., 2005) to approximate density estimation further and thus accelerate the annotation process. When using the EMD kernel density estimator the high computational cost is further compounded by the super-cubic complexity of calculating EMD values between signatures. Several methods have been proposed for reducing the computational cost of kernel smoothing, for a good overview see Girolami and He (2003). In the same article, the authors propose using a proportion of the original training set. Their approach is to finding a small set of kernels, each of which – when adequately weighted – represents the cumulative density of a large group of kernels in the original training set. They define a 'reduced set density estimator', which assigns a weight to each kernel in the training sample $x^{(1)}, \dots, x^{(n)}$:

$$\hat{f}(x, h; z) = \frac{1}{C} \sum_{i=1}^{n} z_i k(x, x^{(i)}; h), \tag{3.8}$$

where $\sum_i z_i = 1$. Their goal is to find a parameter vector z that approximates $\hat{f}(x, h)$ well with as few non-zero weight elements as possible, thereby reducing the amount of computation needed for calculating density. The solution is obtained by minimising the integrated squared error between $\hat{f}(x, h; z)$ and $\hat{f}(x, h)$. The objective function to be minimised is:

$$\hat{z} = \arg\min_{z} \int_{x \in \mathbb{R}^d} |f(x) - \hat{f}(x, h; z)|^2 dx \tag{3.9}$$

The problem can be transformed into

$$\hat{z} = \arg\min_{z} \sum_{i,j}^{N} z_i z_j \int_{x \in \mathbb{R}^d} k(x, x^{(i)}; h) k(x, x^{(j)}; h) dx - 2 \sum_{i}^{N} z_i \hat{f}(x^{(i)}, h), \tag{3.10}$$

and defining the matrix

$$C(i,j) = \int_{x \in \mathbb{R}^d} k(x, x^{(i)}; h) k(x, x^{(j)}; h) dx \tag{3.11}$$

and vector

$$p(i) = \hat{f}(x^{(i)}, h) \tag{3.12}$$

the problem then takes the form

$$\hat{z} = \arg\min_{z} \frac{1}{2} z^{\mathsf{T}} C z - z^{\mathsf{T}} p \tag{3.13}$$

subject to $\sum_i z_i = 1$ and $z_i \geq 0 \; \forall \; i$. Matrix C can be computed exactly when the kernel $k(x, x^{(i)}; h)$ is a Gaussian (Equation 3.3). For other kernels each entry of the matrix can be found by numerically integrating over the data sample. Equation 3.13 is a quadratic programming problem and can be solved using a number of different optimisation algorithms. The optimisation problem and its constraints are such that many of the z terms will be set to zero, while, at the same time, weights which are non-zero are driven to be assigned to areas of high density. This effectively selects a subset of basis points from high-density regions in the data sample.

This is an attractive framework for reducing the computational cost of kernel density estimation, as the only parameter that requires tuning is the kernel bandwidth h. In the case of the EMD kernel, however, it is not obvious how to numerically approximate $C(i,j)$, as the calculation is no longer in a real-valued vector space. However, inspired by the problem formulation that Girolami and He put forward, we propose a simpler way to weight a small number of basis points to approximate the full density estimate. Instead of minimising the integrated squared error between $\hat{f}(x,h)$ and $\hat{f}(x,h;z)$ we obtain the weight vector z by solving

$$\hat{z} = \arg\min_{z} ||p - p_z||^q \tag{3.14}$$

where $p_z(i) = \hat{f}(x^{(i)}, h; z)$ and q is the vector norm, subject to the constraint that only m weights can take nonzero values. The goal of the above equation is to minimise the difference between the full kernel density estimates at the training points and such estimates made using a subset of the original kernels. This subset should be of size m and each kernel should be appropriately weighted to minimise this difference. Define kernel matrix $K(i,j) = k(x^{(i)}, x^{(j)}; h)$, then $p = K i_N$, where i_N is a $N \times 1$ vector whose elements are all $\frac{1}{N}$. In other words p is obtained by averaging K's rows. It follows that one way to solve the problem defined in Equation 3.14 is to quantise K row-wise by picking m prototype rows for the quantisation table, which minimise the quantisation error with respect to the norm q. A clustering algorithm, such as k-means would be suitable for this. This approach will work with any type of kernel, as the kernel matrix K is all that the algorithm requires. However, for large samples the associated dimensionality of K's rows may prevent effective clustering. We sidestep this problem by assuming that points in the training sample that lie near each other will have similar kernel value rows in K, and thus we cluster the points directly, instead of clustering the rows. This shortcut, however, forces us to use the k-medoids algorithm instead of k-means, as the former relies only on the inter-point distances of the data sample, which is all that is available to us in the space induced by the EMD metric. Our procedure then simplifies to the following steps:

- Cluster the training sample $x^{(1)}, \ldots, x^{(n)}$ into m clusters using k-medoids.
- For every basis point $x^{(i)}$ that is a medoid of a cluster, set its kernel weight z_i to the fraction of the training sample falling into that cluster.
- Set the kernel weights of all other basis points to zero.

Clustering points in EMD metric space. The k-medoids procedure for points in EMD metric space is simple:

- Pick m medoid basis points.
- Repeat:
  - Assign each basis point to the cluster of its nearest medoid.
  - For each cluster, select a new medoid that has the lowest average EMD to other points in that cluster.
- until the selected medoids do not change.

The clusters produced by k-medoids are sensitive to the initial choice of medoids. To choose appropriate medoid points we use a modification of the Multiscale Data Condensation algorithm proposed by Mitra et al. (2002). In this method, data points are clustered using hyper-discs of varying radii which are dependent on the density of the data in the region being considered. Specifically, given a sample of data $x^{(1)}, \ldots, x^{(n)}$, a positive integer k and a positive real v:

- For each point $x^{(i)}$, calculate the distance of the kth nearest neighbour of $x^{(i)}$ in the sample. Denote it by $r_{k;x^{(i)}}$.
- Repeat:

  - Select the point $x^{(j)}$ having the lowest value of $r_{k;x^{(j)}}$ and place it in the reduced set E. This point occupies the region of highest density of the current sample.
  - Remove all points from the current sample that lie within a disc of radius $vr_{k;x^{(j)}}$ centered at $x^{(j)}$. Since $r_{k;x^{(j)}}$ is inversely proportional to the estimate of the probability density at $x^{(j)}$, regions of higher probability density are covered by smaller discs, and sparser regions are covered by larger discs. Consequently, more points are selected from the regions having higher density.

- until the current sample is empty.

The number of points selected into the set E can be controlled by varying parameters v and k. The points in E are the m initial medoids supplied to the k-medoids algorithm above.

It should be noted that the above approach applies equally well to data in real vector spaces if we replace EMD with a regular distance metric.

Other techniques to accelerate density estimation. Gaussian mixture models are very popular for approximating nonparametric density functions. One can view each multivariate Gaussian component as approximating the relevant part of the density function, and the overall density function is represented by a linear, weighted combination of these components. Typically component parameters are estimated in terms of sufficient statistics using the Expectation-Maximisation algorithm. Designing such an approach for modelling data in EMD metric space is not straight forward, as it is unclear how to analytically maximise likelihood of probabilistic models in this space.

3.1.2 Features

As mentioned earlier in this article, the effect of 'curse of dimensionality' within our probabilistic annotation framework motivates us to seek the simplest possible features so that reliable density estimation can be performed. We investigate two such feature domains that have been previously shown to correlate with human perception of images — global colour and global texture. Additionally we investigate the use of a more complex feature extraction framework — Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) — the use of which has been recently proposed for object class recognition and automated image annotation.

A case for global colour and texture features. While in this article we strive to extract simple image features primarily for reasons of statistical feasibility, there has been a body of research into identifying the simplest forms of visual stimuli that play a significant role in human visual cognition. Oliva and Schyns (2000) explore the the structure of color cues that allows quick recognition of image scene types. They conclude that colored blobs at a coarse spatial scale can mediate the recognition of scenes without prior recognition of their objects. As mentioned in Section 2.2.2, Torralba and Oliva (2003) show that global frequency domain information of an image — a property closely related to human perception of texture — is highly indicative of its scene category. In the same paper the authors argue that image scene category is a reliable predictor of object presence within the image, owing to strong dependence of the latter upon the former in real world situations.

On this basis we choose global colour and texture features for modelling image keyword distributions. This is unlike many previous approaches that rely on image segmentation or on partitioning of images into blocks (Duygulu et al., 2002; Jeon et al., 2003; Lavrenko et al., 2003; Feng et al., 2004; Carneiro and Vasconcelos, 2005). However, we acknowledge that when performing automated image annotation we rely only on scene information and infer possible relevance of object-related keywords indirectly, based on Torralba and Oliva's hypothesis, outlined in Section 2.2.2. This hypothesis can be demonstrated through some example pictures returned by Flickr[1] search engine in response to the query '*cathedral architecture*' (Figure 3.1). Each is of a different cathedral taken by a different photographer, yet all have two simple features in common – colour layout: white or blue sky on the left and right sides, brown in the middle; and texture layout: smooth sky on the sides, and edges directed towards a similar vanishing point in the middle. Thus the scene keywords for these images could be 'architecture', 'upwards perspective', however many images labelled with 'cathedral' will also share these properties. Pictures in Figure 3.2 — also results of this query — illustrate limitations of this approach, as pointed out by Enser et al. (2005), and discussed in Section 2.2.1.

Global colour

We attempt to model colour properties of images using CIE*Lab* colour space, which was designed to be consistent with human perception of colour similarity. We consider a number of ways to encode the colour distribution of an image into a compact representation: colour moments, colour histograms and colour signatures.

Colour moments are perhaps among the simplest properties one can sample from an image. For each pixel in the image, we compute CIE*Lab* colour values. For each channel, the mean, second, third and fourth central moments are computed resulting in a 12-dimensional feature vector combining colour and texture. We refer to this feature as MargCIE in our experiments. We also designed a tiled image feature to investigate whether performance can be gained by looking at spatial configuration of colour properties. Each image is split into $3 \times 3 = 9$ equal rectangular tiles; within each tile the mean and the second moment are computed for each of the above 3 channels. This results in a 54-dimensional feature vector and we denote this feature as MargCIE-T-3×3.

Colour histograms are an alternative way to represent the global colour distribution of an image. We regularly quantise the 3-dimensional CIE*Lab* space into $m \times m \times m$ three-dimensional bins, denoting this

[1] http://www.flickr.com/

feature as CIE-$m \times m \times m$.

Finally we extracted colour signatures for modelling images using colour distributions in EMD metric space. The CIE*Lab* pixel values in each image were grouped into m clusters using k-means. We refer to this feature as CIE-S-m. The ground distance used for computing the EMD between pairs of such signatures is the Euclidean (L2) distance.

Global texture

We use two different models to represent texture: one proposed by Tamura (1978) and adapted for image retrieval by Howarth and Rüger (2004); and Log-Gabor filters proposed by Field (1987) and implemented by Kovesi (2003). Tamura discovered a number of attributes which humans consider important when characterising texture, and proposed ways to compute them numerically. We extract his *coarseness*, *contrast* and *directionality* properties for each image pixel using a sliding window. As above, we compute four central moments of each of these properties for the entire image, and two central moments for each property in the 9 rectangular image tiles. This results in a 12-dimensional feature space we refer to as MargTamura, and a 54-dimensional space we term MargTamura-T-3×3, respectively.

An alternative way of analysing texture in images is by looking at the distribution of frequencies of the image signal in the intensity domain. Torralba and Oliva use this information in Torralba and Oliva (2003) to categorise images into different scene types. Specifically, they apply Fourier transform to images and find that images within scene categories share frequency components that are characteristic in terms of their scale and orientation. They apply Principal Component Analysis and Linear Discriminant Analysis to the Fourier transform of images to create a real-valued vector space, within which different scene categories are then modelled using Gaussian mixture models.

Differently to their approach, we decompose images into their dominant frequency components by applying 2-dimensional Log-Gabor fiters, which enable the extraction localised frequency information. A standard 2D Gabor filter preserves a part of the image signal in the spatial domain that corresponds to the signal's frequency components within a particular range of scales and orientations. Such filtering is achieved by multiplying the 2D Fourier transform of the image with a 2D Gaussian function, the width and orientation of which specifies the range of frequencies preserved in the signal. A Gabor filter bank is a set of filters, which, together, cover the entire frequency spectrum of an image. By applying each Gabor filter to the image separately one can create a multichannel image representation, where each original pixel is described by a vector of the filters' response values at the pixel's location. Turner first implemented such a filter bank to analyse image texture in Turner (1986). A Log-Gabor filter, proposed by Field (1987), is similar to its regular counterpart but uses a function corresponding to a Gaussian in the log space for filtering the signal. Field's studies suggested that such filters are more adequate for encoding frequencies found in natural images.

We use the Log-Gabor filter bank designed by Kovesi (2003) to extract frequency information from images, which partitions the frequency spectrum of an image using filters ranging across four scales and six orientations. We design two different features from the resulting 24-channel image representation. The first is a 48-dimensional real-valued feature vector, where each filter output is encoded by its mean and its standard deviation across the entire image. We denote this feature Log-Gabor. The second is a signature of joint filter responses, obtained as follows:

- partition the 24-channel image representation into $n \times n$ blocks
- create a 24-dimensional feature vector for each block by averaging the response of each filter within the block
- cluster block feature vectors into m clusters using k-means to obtain a signature of joint filter responses.

The above feature is then used to model images using joint frequency distributions within EMD metric space. As the underlying feature space is 24-dimensional, it is clear that the alternative of creating a regularly quantised histogram of joint frequencies would be prohibitive (a quantisation scheme with just two bins per channel would result in over 16 million feature vector components). Partitioning images into blocks and averaging filter responses is intended to make clustering more efficient. We abbreviate this feature as Log-Gabor-S-m. The ground distance used for computing the EMD between pairs of such signatures is the Manhattan (L1) distance.

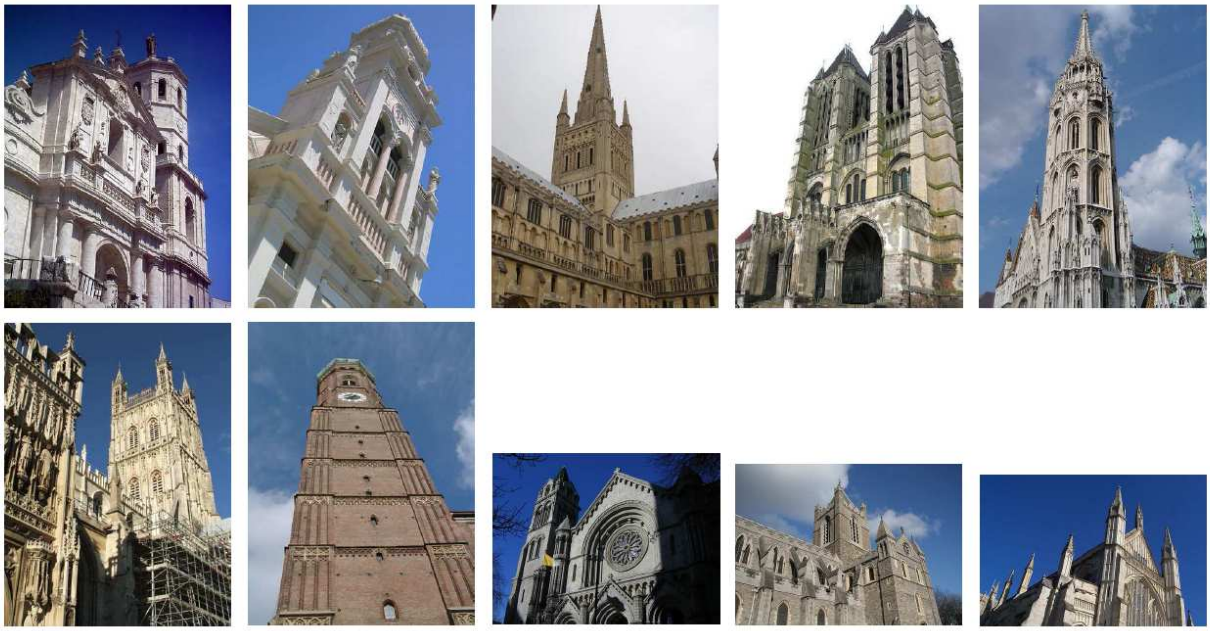


Figure 3.1: Different images of cathedrals that are visually similar

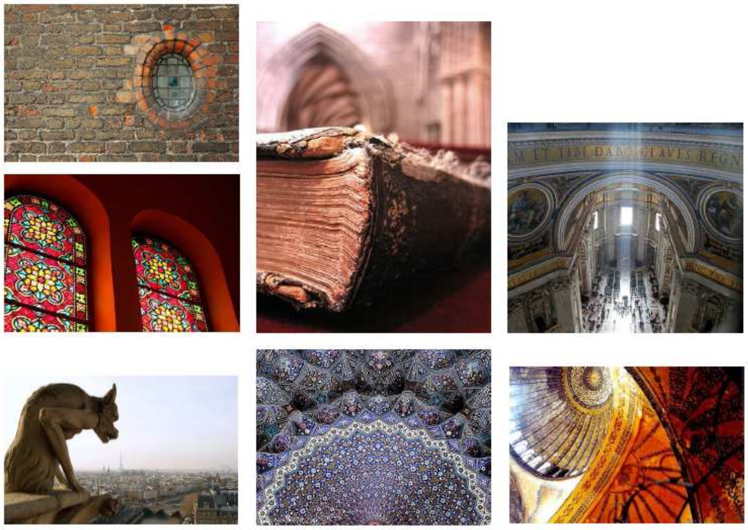


Figure 3.2: Different images of cathedrals that are visually dissimilar

Scale Invariant Feature Transform

Interest region extraction has been investigated as an alternative to employing image segmentation or global image features for content-based image description. The general idea is to extract regions of the image that are in some way characteristic of the type of the depicted object or scene, and performing analysis on those instead of doing so on the entire image (the reader is referred to Sebe et al. (2003) for a good overview). Recently, the Scale Invariant Feature Transform (SIFT) descriptor has become a popular technique for locating and characterising such regions. The intended purpose of this descriptor is to extract unique interest regions, or *keypoints*, of particular objects within images, such that these keypoints are robust to a number of image alterations. Initial applications of this feature included locating copies of example objects and scenes within image databases and inside video streams, and automatically aligning (or ‘stitching’) multiple photographs of the same scene into a panoramic picture by identifying shared keypoints. The feature is extracted as follows:

- the black and white version of the image is progressively Gaussian blurred resulting in a series of images blurred at different scales
- these images are subtracted from their neighbours in the series to produce a new series of images (‘difference-of-Gaussians’ images at different scales)
- extremal points are located at each scale: each pixel in the image is compared to its 8 neighbours and the 9 pixels each (corresponding to the pixel and the 8 neighbours) at images at neighbouring two scales
- keypoints are chosen from these extrema

- for each keypoint, histograms of gradient directions in the 16x16-pixel surrounding window: the dominant direction specifies the keypoint's orientation
- a 128-dimensional vector is generated by splitting the surrounding window into a 4×4 grid, and the gradient direction into 8 bins within each grid square; rotation invariance is achieved by aligning the gradient histograms with the keypoint's orientation

The extrema localisation in SIFT favours points with abrupt intensity changes, such as corners and sharp edges, often pertaining to the more textured parts of the image. A typical natural image will have between a hundred and a few thousand of such keypoints, depending on its size. Copies of objects and scenes can be efficiently detected and spatially aligned by identifying a small subset of matching keypoints, owing to the latter's large degree of invariance to rotation, scale and illumination conditions (Lowe, 2004). However, intuitively it would appear to be difficult to build generalised models of object and scene categories using keypoints that are so specific to visual attributes of particular images. The suitability of SIFT for object categorisation and object retrieval has been explored by Csurka et al. (2004) and Sivic and Zisserman (2003), respectively, and that for automated image annotation has been investigated by Hare et al. (2006). These methods rely on transforming SIFT features into discrete tokens by vector quantizating the keypoints found in the training set.

Csurka et al. (2004) obtain the quantisation scheme by grouping the SIFT keypoints into one thousand clusters using k-means. Each image is then represented by a set of histogram bins, counting how many of the image's keypoints fall into each particular cluster. An object classifier is then constructed based on this feature representation, and the authors report good classification performance on a dataset with 7 object classes. In a similar manner, Sivic and Zisserman (2003) use quantised SIFT descriptors to imitate words in a text document, and apply traditional text retrieval techniques for matching exemplar objects with potentially relevant keyframes in video collections. Hare et al. (2006) used this text-like image representation for automated image annotation by treating the quantised SIFT keyponts as visual words and applying an information retrieval model called 'cross-language latent semantic indexing' to translate the content of an image into its predicted captions.

In this article we explore a feature representation based on SIFT keypoints. For each image we generate a SIFT signature by clustering the image's keypoint feature vectors into m clusters with k-means. These signatures are then used in conjunction with the EMD kernel density estimation. Using this representation, which we term SIFT-S-m, we compare SIFT features with the much simpler global features described earlier in this section.

3.2 Experimental results

3.2.1 Image data

Corel dataset. One of the datasets we use is the one by Duygulu et al. (2002). To make our results comparable to those recently published in Duygulu et al. (2002); Jeon et al. (2003); Lavrenko et al. (2003) we use the same training and test dataset partition as in Duygulu et al. (2002), where there are 4,500 training images and 500 test images. To optimise the kernel bandwidth parameters for different features we randomly divide the training set into 3,800 training images and 700 images on which different bandwidth settings are evaluated.

A possible problem with the above dataset, however, is that it has a very small test set of just 500 images. For our experiments we define a 'realistic' collection as one that contains 10,000 or more images and in which images come from a variety of sources. We have compiled two such datasets ourselves – *Getty* and *Web images* – which, we believe, reflect two different realistic image retrieval scenarios.

Getty dataset. We attempted to build a collection of images which are similar to those stored and distributed by commercial image archives. For this we downloaded 20,000 medium-resolution thumbnails of photographs from the Getty Image Archive website[2], together with the annotations assigned by the Getty staff to catalogue those pictures. The photographs were obtained by submitting the following two queries to the Getty website – "*photography, image,* not *composite,* not *enhancement,* not *'studio setting',* not *people*" (18,796 images) and "*photography, image, people,* not *composite,* not *enhancement,* not *'studio setting'*" (1,204 images) – both queries having the additional search option to exclude illustrations. With these queries we sought to obtain a random selection of photos, which excludes any non-photographic content, any digitally composed or enhanced photos and any photos taken in unrealistic studio settings. The constraint to exclude people in the majority of the photographs is imposed to reduce the semantic ambiguity of annotations. The resulting dataset contains pictures from a number of different photo vendors, which – we hope – reduces the chance of unrealistic correlations between keywords and image contents.

Annotations for Getty images come in three different kinds: subjects (e.g. '*tiger*'), concepts (e.g. '*emptiness*') and styles (e.g. '*panoramic photograph*'). We created our vocabulary using subject keywords only, of which there were over 10,000. We restricted the range of keywords to those, which occur in fewer than 10% of the images and those, which occur more than 50 times. We then pruned references to specific locations (e.g. '*europe*', '*japan*'), descriptions of dominant image colour, verbs and abstract nouns (e.g. '*flying*', '*close-up*'). This resulted in a final list of 247 words ranging from specific objects (e.g. '*insect*', '*church*') to more general object categories (e.g. '*building structure*') and scene properties (e.g. '*urban scene*', '*autumn*). We randomly split the dataset into training and test partitions of equal size. The

[2] http://creative.gettyimages.com

training set is further randomly split into another training and validation sets of equal size for estimating the bandwidth parameter value for different feature combinations.

Web images. We constructed a different dataset to measure the effectiveness of our approach on images that can typically be found on the World Wide Web. For this we obtained a set of images from the PicSearch[3] search engine with the 11 following queries that define the dataset vocabulary:
aerial view, building exterior, clouds, crowd, flower, grazing animal, jug, mountain, mugshot, sunset, underwater fish

We manually filtered out irrelevant images from each query result, leaving between 400 and 1,200 images per query and 8,714 images in total. We then used the ESP dataset (Ahn and Dabbish, 2004) to obtain a representative sample of the great variety of images available on the web that are not captured by the above categories[4]. From this dataset we excluded images with annotations related to any of our categories leaving 61,100 images which we label as 'nonrelevant'. Images downloaded from PicSearch were aggregated and randomly split into equal training and test partitions and the remaining ESP images were randomly partitioned such that the training and the test partitions have, respectively, 10,000 and 59,814 images in total. Partitioning the collection in this way reflects the fact that most of the images on the World Wide Web are irrelevant to any given user query. The training set is likewise randomly split into another training and validation sets of equal size for estimating optimal bandwidth value for each set of features.

COIL-100. We use the Columbia Object Image Library (Nene et al., 1996) to verify that our method of handling SIFT features is adequate. This is a collection of images of 100 objects set against a black background. Each object was rotated 360 degrees and images were taken at 5 degree intervals, resulting 72 poses per object. Illumination was kept constant during this process. We split the collection such that there are 5 random poses per object in the training set, and 2,500 random images in the test set. Since SIFT is a feature extraction technique that was originally designed for invariant object matching, we can use this experimental set-up to assess whether the SIFT-S-m signature we define in Section 3.1.2 preserves enough information from original keypoints.

Details of these collections (image identifiers and their keywords) are available online[5].

3.2.2 Performance evaluation

We use the same experimental setup as in Jeon et al. (2003) and evaluate annotation accuracy by looking at ranked retrieval performance. In this setup, the manually annotated image collection is split into training and test sets. Keyword models are estimated on the training set and images in the test set

[3] http://www.picsearch.com

[4] http://hunch.net/~learning/ESP-ImageSet.tar.gz

[5] http://www.beholdsearch.com/research/datasets/

are assumed to be unlabelled. Given an m keyword query, an image in the test collection is treated as relevant at evaluation time if it contains all of the query's keywords in its set of annotations.

For the Corel dataset all 1– 2– and 3-word queries are generated that would yield at least 2 relevant images in the test set. For the Getty dataset we require at least 10 relevant images for any given query (to cut down the greater number of queries due to the larger size of the test set), and generated all possible 1–4 word queries under this constraint. In Web image and COIL-100 datasets keywords do not co-occur so only single keyword queries are appropriate. Given an m-word query $Q = \{q_1, q_2, \ldots, q_m\}$ the retrieval score for an image x is defined as

$$p(q_1, q_2, \ldots, q_m|x) = \prod_{i=1}^{m} p(q_i|x). \tag{3.15}$$

Query results are then evaluated using the standard average precision metric described in Section 2.2.4.

For the Corel and Web images the image block size used by the Log-Gabor signature is 6×6 pixels. For Getty and COIL collections the sizes are 7×7 and 4×4 pixels, respectively.

Benchmark results are reported in Table 3.1. It is important to note that the experimental setup of Lavrenko et al. (2003) is different to that of Feng et al. (2004) and Carneiro and Vasconcelos (2005). The former generate single word queries that would yield at least 2 relevant images in the test set, whereas the latter require only 1 relevant document per query. In this chapter we shall carry out detailed comparisons against the results of Lavrenko et al. (2003) since the authors use the same setup as Jeon et al. (2003) that we originally used for our experiments. However, we shall also make separate comparisons with the results reported in Feng et al. (2004) and Carneiro and Vasconcelos (2005).

Single feature results

In this section we use the Corel dataset to find good real-valued vector features of colour and texture for subsequent use. The comparison of features is based on retrieval precision of single-word queries evaluated on the withheld portion of the dataset. We also compare keyword retrieval results of individual features against those of standard annotation models reported in Lavrenko et al. (2003); Feng et al. (2004); Carneiro and Vasconcelos (2005) and retrieval at random, shown in Table 3.1. Tables 3.2 and 3.3 show retrieval precision for automated annotation performed in different feature spaces using Laplace and Gaussian kernel density estimation, respectively. In all cases using the Laplace kernel results in better performance. The most significant difference can be noted for the high-dimensional colour histogram features. Following this result we choose the Laplace kernel for this and all subsequent experiments involving density estimation in real vector spaces.

When using the Laplace kernel, MargCIE moment feature is competitive with CIE-5×5×5 and CIE-8×8×8 histogram features, though the latter have a much higher dimensionality. The locally-sensitive

Model	1 word	2 words	3 words
Continuous Relevance Model (Lavrenko et al., 2003)	0.2353	0.2534	0.3152
Multiple Bernoulli Relevance Model (Feng et al., 2004)	0.30	—	—
Hierarchical Mixtures (Carneiro and Vasconcelos, 2005)	0.31	—	—
Random	0.0325	0.0206	0.0175

Table 3.1: Performance benchmarks for the Corel dataset. Figures are shown as originally reported.

.

Feature	1 word	2 words	3 words	1 word (withheld)
CIE-8×8×8	0.2007	0.1743	0.1964	0.1734
CIE-5×5×5	0.2087	0.1996	0.2274	0.1697
MargCIE-T-3×3	0.2956	0.2976	0.3536	0.2492
MargCIE	0.2192	0.2172	0.2630	0.1966
Tamura-T-3×3	0.1562	0.1534	0.1791	0.1461
MargTamura-T-3×3	0.1592	0.1537	0.1825	0.1267
MargTamura	0.1052	0.0918	0.1112	0.0936
Log-Gabor	0.1924	0.1857	0.2305	0.1725

Table 3.2: Ranked retrieval performance on Corel dataset: automated annotation using the Laplace kernel density estimation (k_L). Features that perform well on the withheld set are highlighted in bold.

MargCIE-T-3×3 has the best performance – competitive on its own with the benchmark results. Colour appears to be more important than texture for this particular dataset. Locally sensitive Tamura feature moments (MargTamura-T-3×3) performs better than its entire-image counterpart, however the best real-valued vector texture feature turns out to be Log-Gabor.

Signature size selection

When automatically annotating images it is interesting to consider how much feature information must be preserved to achieve good retrieval accuracy. We investigate this by looking at how varying the number of clusters per image signature affects automated annotation using EMD kernel density estimation. A signature that consists of fewer clusters can be said to compress the underlying multidimensional distribution more, retaining fewer of its salient characteristics. Figures 3.3, 3.4 and 3.5 show how single

Feature	1 word	2 words	3 words	1 word (withheld)
CIE-8×8×8	0.1532	0.1385	0.1608	0.1275
CIE-5×5×5	0.1510	0.1359	0.1499	0.1315
MargCIE-T-3×3	0.2692	0.2859	0.3370	0.2245
MargCIE	0.2054	0.2008	0.2252	0.1790
Tamura-T-3×3	0.1437	0.1387	0.1687	0.1234
MargTamura-T-3×3	0.1405	0.1336	0.1550	0.1072
MargTamura	0.0981	0.0873	0.1041	0.0872
Log-Gabor	0.1816	0.1792	0.2173	0.1627

Table 3.3: Ranked retrieval performance on Corel dataset: automated annotation using the Gaussian kernel density estimation (k_G)

keyword retrieval precision on the Corel dataset varies with signature size for CIE*Lab* colour, Log-Gabor wavelet and SIFT features, respectively.

Remarkably, one obtains mean average precision of 0.1909 using colour signatures with just 2 colour clusters per image. Good colour signature size appears to be 16 clusters, resulting in retrieval performance that is competitive with the benchmark result of Lavrenko et al. (2003)f reported in Table 3.1. It is also worth noting that using the CIE*Lab* feature within EMD kernel density estimation framework results in significantly better performance than modelling densities of CIE*Lab* histograms in real vector space. Similarly, using signatures consisting of just 16 clusters of Log-Gabor wavelet responses achieves precision that is competitive with Continuous Relevance Model and outperforms the real-valued vector version of the Log-Gabor feature – an impressive result considering that colour information is not utilised. Although increasing the number of clusters to 24 improves results further, we do not choose it because it comes at a much higher computational cost when calculating EMD. These results indicate that by using the EMD kernel one can model densities of multidimensional image feature distributions with increased robustness. On the other hand, using the SIFT signature appears to underperform on the Corel dataset regardless of the signature size. We note this difference in performance between the global features and our SIFT representation, and discuss it in greater depth later in this section. Table 3.4 shows retrieval scores of 1-, 2- and 3- word queries when the above features are represented by 16-cluster signatures.

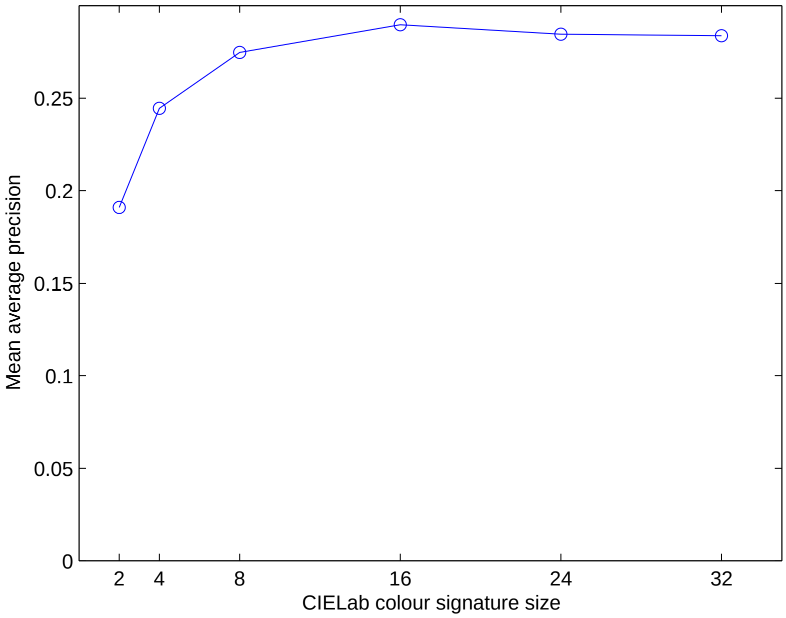


Figure 3.3: Effects of colour signature size on the retrieval accuracy for the Berkeley Corel collection

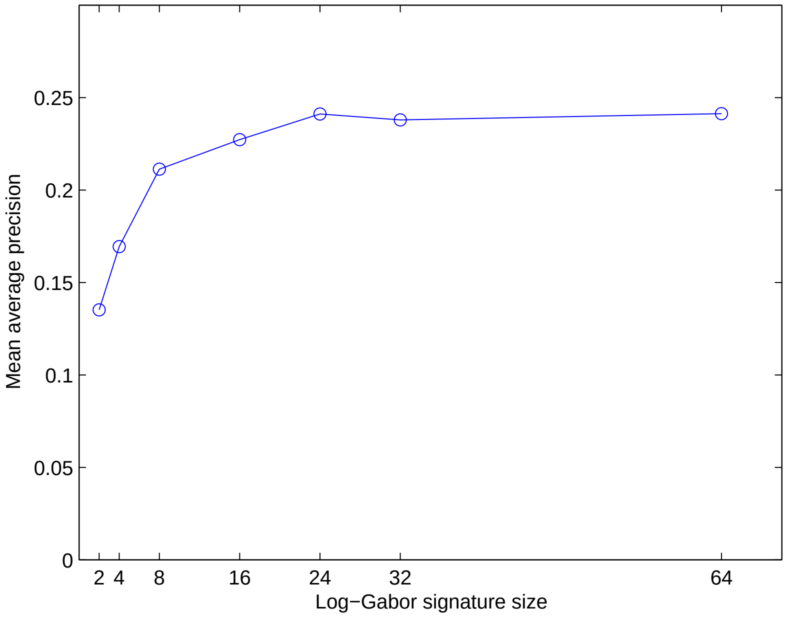


Figure 3.4: Effects of wavelet signature size on the retrieval accuracy for the Berkeley Corel collection

Feature	1 word	2 words	3 words	1 word (withheld)
CIE-S-16	0.2897	0.2940	0.3344	0.2497
Log-Gabor-S-16	0.2273	0.2095	0.2436	0.2152
SIFT-S-16	0.1270	0.1071	0.1190	0.1161

Table 3.4: Ranked retrieval performance on Corel dataset: automated annotation using the EMD kernel density estimation (k_L)

Feature combination

In this subsection we combine features which performed well on the withheld set of Corel images individually. We evaluate these feature combinations on all three datasets. In the case of EMD kernel density estimation, features F_1 and F_2 are combined by adding EMD values between signatures under each feature:

$$k_E(s, s^{(i)}; h) = \frac{1}{h} \exp\left(-\frac{d\left(s_{F_1}, s_{F_1}^{(i)}\right) + d\left(s_{F_2}, s_{F_2}^{(i)}\right)}{h}\right) \tag{3.16}$$

We note that this is a heuristic way to combine features within the EMD kernel. It assumes that the individual distributions of the features that are to be combined are entirely independent. A more principled approach would be to compute joint distributions of these features and calculate the EMD values between them.

For real-valued vector features we simply concatenate respective feature vectors for each image prior to density estimation. We selected three vector features – MargCIE-T-3×3, Tamura-T-3×3 and Log-

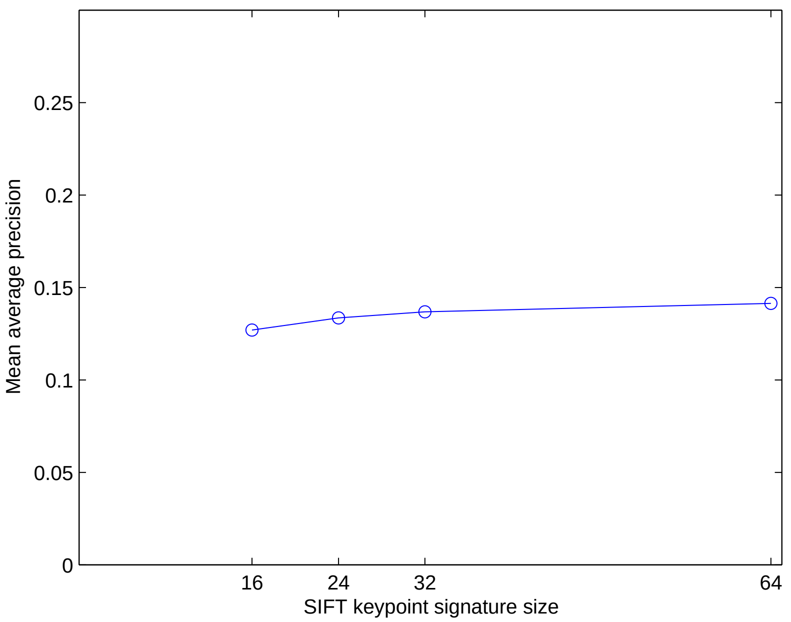


Figure 3.5: Effects of SIFT signature size on the retrieval accuracy for the Berkeley Corel collection

Gabor – which perform well on the Corel dataset individually, as shown in Table 3.2. Table 3.6 shows feature combination results on the Corel dataset. All vector feature combination schemes perform better than the benchmark result of Lavrenko et al. (2003): the best result is achieved by combining CIE-S-24 and Log-Gabor-S-24 signatures. Combining CIE-S-16 and Log-Gabor-S-16 under EMD kernel density estimation results in similar accuracy, but is much cheaper computationally. Combining MargCIE-T-3×3, Tamura-T-3×3 and Log-Gabor falls slightly below the signature combinations, even though it uses positional information. We choose the CIE-S-16+Log-Gabor-S-16 and MargCIE-T-3×3+Tamura-T-3×3+Log-Gabor combinations and apply them to the Getty and Web image datasets. We refer to these feature combinations as ‘combined vector’ and ’combined signature’, respectively.

At this point let us compare results of these feature combinations to those of Feng et al. (2004) and Carneiro and Vasconcelos (2005) shown in Table 3.1. The results of this comparison, presented in Table 3.5, indicate that our approach rivals their benchmark figures. Interestingly, by adding the MargCIE feature to the combined vector feature one can obtain the mean average precision of 0.32, thus outperforming these benchmark results. However, this increase in precision does not turn out to be statistically significant.

Table 3.7 shows that for single-word queries, the precision gap between random retrieval and the selected feature combinations for the Getty dataset is much smaller than that for the Corel dataset. This indicates that low-level image feature distributions for the Getty keywords are less accurate. One

Feature	1-word query M.A.P.
Combined signature	0.31
Combined vector	0.31

Table 3.5: Results obtained using the experimental setup of Feng et al. (2004); Carneiro and Vasconcelos (2005)

Kernel and feature	1 word	2 words	3 words	1 word (withheld)
k_E CIE-S-16+Log-Gabor-S-16	0.3511	0.3402	0.3852	0.3080
k_E CIE-S-24+Log-Gabor-S-24	0.3558	0.3453	0.3959	0.3126
k_L MargCIE-T-3×3+Tamura-T-3×3	0.3211	0.3279	0.3840	0.2805
k_L MargCIE-T-3×3+Log-Gabor	0.3349	0.3329	0.3891	0.2982
k_L MargCIE-T-3×3+Tamura-T-3×3+Log-Gabor	0.3409	0.3342	0.3791	0.3004

Table 3.6: Feature combination evaluation for ranked retrieval on the Corel dataset

can see from results in Table 3.8 that for the Web image dataset this difference is much more convincing, which shows that the low-level features are more appropriate. For both datasets it is evident that colour information is important, regardless of the density estimation technique. Table 3.8 also shows that SIFT-S-16 feature performs substantially worse than the simple global features on the Web dataset, much like in the case of Corel image data.

More on EMD kernel density estimation performance

As noted in Section 3.2.2, using the CIE*Lab* feature for EMD kernel density estimation on the Corel dataset results in better accuracy than modelling densities of CIE*Lab* histograms in real vector space. Better results are also obtained on the same dataset when Log-Gabor-S-16 signature feature is used instead of its real-vector counterpart, Log-Gabor. This also turns out to be true for the Getty dataset, as shown in Table 3.9. The same table shows that for the Web image collection this is true for the CIE*Lab* colour features, but Log-Gabor performance is about the same for both density estimation techniques.

Data reduction

In this section we assess the data reduction technique outlined in Section 3.1.1. Since we are using non-parametric density estimation, reduction in the amount of training data will correspond to the reduction in annotation time. We filtered the vocabularies of Corel and Getty datasets such that each keyword has

Feature	1 word	2 words	3 words	4 words
Combined vector	0.0955	0.0522	0.0608	0.0865
Combined signature	0.0918	0.0539	0.0669	0.0964
Log-Gabor	0.0451	0.0165	0.0129	0.0123
Log-Gabor-S-16	0.0576	0.0248	0.0223	0.0251
Random	0.0142	0.0034	0.0025	0.0023

Table 3.7: Mean average precision for ranked retrieval on the Getty dataset

Keyword	Combined vector	Combined signature	Log-Gabor	Log-Gabor-S-16	SIFT-S-16
Aerial view	0.3433	0.2966	0.1713	0.1502	0.1329
Building exterior	0.2553	0.1924	0.1110	0.0935	0.1115
Flower	0.2023	0.1731	0.0948	0.1356	0.0480
Jug	0.3636	0.2178	0.1441	0.1128	0.0335
Mugshot	0.4449	0.3294	0.1622	0.2176	0.0820
Clouds	0.3894	0.3067	0.2503	0.1965	0.0871
Crowd	0.1874	0.1659	0.0819	0.0697	0.1064
Grazing animal	0.2632	0.2207	0.0754	0.1243	0.0614
Mountain	0.3735	0.3748	0.1679	0.1600	0.0824
Sunset	0.6150	0.5768	0.3094	0.3005	0.0991
Underwater fish	0.2373	0.1972	0.0985	0.0971	0.0840
Average	0.3341	0.2774	0.1515	0.1507	0.0844

Table 3.8: Mean average precision for ranked retrieval on the Web image dataset (random retrieval mean average precision = 0.0074)

Collection	CIE-5×5×5	CIE-8×8×8	CIE-S-16	Log-Gabor	Log-Gabor-S-16
Getty	0.0423	0.0497	**0.0592**	0.0451	**0.0576**
Web images	0.0543	0.0711	**0.1187**	**0.1515**	0.1507

Table 3.9: Comparison of EMD and Laplace kernel density estimation for single keyword queries

at least 100 associated training images. This allows to observe effects of reducing each keyword training sample to very small percentages of original sample size. For Getty we additionally filtered out keywords with low mean average precision so that the effect of reducing training sample size is more noticeable for each keyword. For the Corel dataset the selected keywords were:

sun, jet, polar, cars, horses, plane, clouds, flowers, garden, plants, field, close-up, tracks, birds, bear, sand, grass, mountain, stone, snow, leaf, boats, bridge, buildings, rocks, sky, ruins, valley, water, statue, people, tree, street, hills, house, beach.

For the Getty dataset the following keywords were selected:

Building Exterior, Cityscape, Clear Sky, Cloud, Dusk, Field, Flower, Fog, Food, Grass, Horizon, Landscape, Leaf, Lush Foliage, Mammal, Mountain, Night, Non-Urban Scene, One Animal, One Person, Plant, River, Sea, Sky, Skyline, Skyscraper, Snow, Sun, Sunset, Tree, Underwater, Urban Scene, Vegetable, Window, Winter, Woods

For both datasets, images which are not labelled with any of the above keywords were assigned the keyword ‘other’, which is explicitly modelled using our approach but is not included when calculating mean average precision. We term these datasets **modified Corel** and **modified Getty**, respectively. The Web dataset is kept intact. Tables 3.13, 3.14 show average precision single-keyword queries for the two modified datasets.

We apply the data reduction technique described in Section 3.1.1 to each individual keyword training sample T_w to obtain a reduced set density estimate $\hat{f}_w(x, h; z)$ for the keyword w. We set the parameter $k = 1$. By varying a single parameter v, defined in Section 3.1.1, for all keywords, we vary the amount of data retained by the data reduction procedure for each keyword density estimate. For each of the three datasets, we report mean average precision against the average percentage of data kept across all keyword density estimates, the specific feature combination used and the value of v. As noted in Section 3.1.1, our data reduction technique can work in real vector spaces as well as in the EMD metric space. In the case of the concatenated vector feature MargCIE-T-3×3+Tamura-T-3×3+Log-Gabor we use the Manhattan distance for Multiscale Data Condensation and k-medoids clustering and use the resulting basis point weights within a Laplace kernel density estimator.

Table 3.10 shows the effects of withholding different amounts of data for EMD and Laplace kernel density estimation on modified Corel dataset annotation performance. Single-word query average precision is used for evaluation. For the reduced Laplace kernel density estimator 83% of original performance is preserved when, on average, 20% of original data per keyword is used; 75% when 5% of the data is used. For the reduced EMD kernel density estimator 79% of performance is achieved through keeping only 15% of the data per keyword on average, and 68% when using 4%. Note, however, that for this particular dataset configuration the average precision for random retrieval is quite high.

We compare our data-reduction method to another popular technique for speeding up density estimation – a Gaussian mixture model (GMM). We use the Netlab software package[6] to fit a GMM to each keyword sample with 10 times fewer components than there were points in that sample. At the same time we use our method in conjunction with Laplace kernel density estimation, and we modified the parameter v so that roughly 10% of the training sample is retained on average for all keywords. We carry out this comparison for the Log-Gabor feature. We chose this low-dimensional feature space to avoid the problem of overfitting for the GMM, and constrained the covariance matrix of each Gaussian to be diagonal to avoid 'singularities' during the fitting process. The two bottom rows of Table 3.10 show that roughly the same performance is obtained for similar data reduction levels. The fact that that the accuracy of our method is competitive with the GMM in real vector space suggests that it is an appropriate technique for accelerating density estimation in EMD metric space, within which GMMs cannot be directly applied.

Tables 3.11 and 3.12 show relative performance of our reduction technique on modified Getty and Web image datasets. Interestingly the accuracy of the reduction technique for the Web image dataset is poor compared to other cases. 32% of mean average precision is preserved with 13% of the data retained for when using the EMD kernel and 52% is kept with 17% of data for the Laplace kernel. This indicates that it is important to retain more original data when estimating keyword densities on this collection.

[6] http://www.ncrg.aston.ac.uk/netlab/

Kernel, feature and reduction parameters	M.A.P.	Fraction of data
k_L Combined vector *full*	0.4753	100.00%
k_L Combined vector *reduced* v=1.3	0.3979	19.32%
k_L Combined vector *reduced* v=2	0.3588	5.34%
k_E Combined signature *full*	0.4776	100.00%
k_E Combined signature *reduced* v=1.3	0.3749	15.29%
k_E Combined signature *reduced* v=2	0.3244	3.59%
Log-Gabor GMM	0.2076	10.00%
k_L Log-Gabor *reduced* v=2	0.2348	10.18%
Random	0.0703	—

Table 3.10: Reduced set density estimation on the modified Corel dataset

Kernel, feature and reduction parameters	M.A.P.	Fraction of data
k_L Combined vector *full*	0.2188	100.00%
k_L Combined vector *reduced* v=1.3	0.1778	20.51%
k_E Combined signature *full*	0.2115	100.00%
k_E Combined signature *reduced* v=1.3	0.1672	16.83%
Random	0.0402	—

Table 3.11: Reduced set density estimation on the modified Getty dataset

Kernel, feature and reduction parameters	M.A.P.	Fraction of data
k_L Combined vector *full*	0.3341	100.00%
k_L Combined vector *reduced* v=1.3	0.1744	16.99%
k_E Combined signature *full*	0.2774	100.00%
k_E Combined signature *reduced* v=1.3	0.0892	13.24%
Random	0.0074	—

Table 3.12: Reduced set density estimation on the Web image dataset

Keyword	Combined vector	Combined signature	Log-Gabor	Log-Gabor-S-16	Random
water	0.5763	0.5657	0.4808	0.4844	0.2217
sky	0.6037	0.6231	0.4409	0.5086	0.1875
tree	0.3691	0.3783	0.3162	0.3198	0.2413
people	0.5018	0.5807	0.3657	0.4170	0.1694
grass	0.4626	0.5104	0.2622	0.3899	0.0993
buildings	0.4514	0.3704	0.3011	0.3815	0.1192
mountain	0.3348	0.3210	0.2022	0.1975	0.1003
flowers	0.4528	0.4451	0.2459	0.3215	0.0513
snow	0.5687	0.5721	0.3324	0.3147	0.0576
clouds	0.3437	0.3749	0.1729	0.2667	0.0624
rocks	0.1631	0.1848	0.1334	0.1043	0.0543
stone	0.5993	0.6830	0.2480	0.3084	0.0396
street	0.4075	0.3947	0.3133	0.3321	0.0471
plane	0.8840	0.7648	0.7176	0.7903	0.0609
field	0.4435	0.5027	0.4537	0.4475	0.0347
bear	0.8144	0.7417	0.6017	0.6346	0.0373
sand	0.3086	0.3588	0.1075	0.1624	0.0886
birds	0.4354	0.4536	0.2313	0.2692	0.0432
beach	0.4167	0.4818	0.1472	0.1775	0.1511
boats	0.1624	0.2118	0.2140	0.1094	0.0322
jet	0.8674	0.8394	0.7608	0.7745	0.0870
leaf	0.3855	0.4406	0.1889	0.2826	0.0267
cars	0.7369	0.7139	0.5552	0.6773	0.0705
plants	0.3245	0.2410	0.1127	0.1467	0.0401
house	0.3632	0.3293	0.1181	0.1109	0.0330
bridge	0.2962	0.1335	0.1728	0.0914	0.0371
valley	0.4884	0.4995	0.2577	0.2383	0.0166
polar	0.9329	0.7885	0.6269	0.6429	0.0218
garden	0.2064	0.2621	0.1297	0.1099	0.0436
hills	0.1746	0.1304	0.0725	0.0759	0.0730
close-up	0.1124	0.2119	0.1474	0.1677	0.0231
ruins	0.2745	0.3792	0.0814	0.1623	0.0235
statue	0.3183	0.3644	0.1328	0.1584	0.0276
tracks	0.9103	0.9377	0.7810	0.9183	0.0214
horses	0.8298	0.8277	0.6302	0.7103	0.0396
sun	0.5913	0.5761	0.3928	0.4661	0.0486
Average	0.4753	0.4776	0.3180	0.3520	0.0703

Table 3.13: Average precision for ranked retrieval on the modified Corel dataset

Keyword	Combined vector	Combined signature	Log-Gabor	Log-Gabor-S-16	Random
Building Exterior	0.2130	0.2091	0.1684	0.1752	0.0740
Cityscape	0.1531	0.1420	0.0737	0.0959	0.0201
Clear Sky	0.3122	0.3364	0.1287	0.1876	0.0466
Cloud	0.2916	0.2765	0.1858	0.2196	0.0762
Dusk	0.1913	0.1787	0.0850	0.0815	0.0443
Field	0.2533	0.2467	0.1108	0.1432	0.0316
Flower	0.1658	0.1763	0.0710	0.0986	0.0317
Fog	0.1436	0.1357	0.0988	0.1120	0.0124
Food	0.4358	0.4039	0.2106	0.2218	0.0389
Grass	0.2215	0.2139	0.0856	0.1076	0.0364
Horizon	0.1670	0.1600	0.1254	0.1396	0.0389
Landscape	0.1448	0.1556	0.0880	0.1018	0.0296
Leaf	0.2110	0.2516	0.0956	0.1121	0.0243
Lush Foliage	0.2596	0.2506	0.0554	0.0747	0.0121
Mammal	0.3227	0.3051	0.1403	0.1699	0.0229
Mountain	0.1970	0.1971	0.1090	0.1519	0.0347
Night	0.3406	0.3498	0.1281	0.1676	0.0661
Non-Urban Scene	0.1741	0.1625	0.1128	0.1158	0.0597
One Animal	0.3482	0.3032	0.1865	0.2242	0.1088
One Person	0.3289	0.2561	0.2100	0.2308	0.0579
Plant	0.1764	0.1739	0.1213	0.1263	0.0450
River	0.1753	0.1689	0.0978	0.1195	0.0227
Sea	0.1861	0.2025	0.1240	0.1377	0.0465
Sky	0.2804	0.2906	0.1409	0.1595	0.0839
Skyline	0.1709	0.1325	0.1323	0.1373	0.0163
Skyscraper	0.1391	0.1385	0.0962	0.0842	0.0217
Snow	0.1861	0.1640	0.0608	0.0677	0.0348
Sun	0.1510	0.1695	0.0471	0.0546	0.0155
Sunset	0.1988	0.2086	0.0799	0.0946	0.0209
Tree	0.2165	0.2133	0.1675	0.1790	0.0718
Underwater	0.2451	0.2649	0.0622	0.0871	0.0230
Urban Scene	0.1912	0.2072	0.1495	0.1716	0.0670
Vegetable	0.1777	0.1623	0.0701	0.0927	0.0159
Window	0.1772	0.1321	0.1196	0.1435	0.0370
Winter	0.1628	0.1180	0.0373	0.0387	0.0271
Woods	0.1686	0.1549	0.0878	0.0906	0.0146
Average	0.2188	0.2115	0.1129	0.1310	0.0397

Table 3.14: Average precision for ranked retrieval on the modified Getty dataset

Feature	M.A.P.
SIFT-S-16	0.8078
SIFT-S-64	0.8903
SIFT-S	0.8915
Log-Gabor-S-16	0.8609
Log-Gabor-S-64	0.8757
Random	0.0100

Table 3.15: SIFT-S-16 and Log-Gabor-S-16 on the COIL-100 dataset

SIFT signature performance

As we have seen above, using SIFT within EMD kernel density estimation framework results in substandard annotation accuracy on the Corel and Web image datasets. We outlined our intuition earlier for a possible reason behind this: that it may be difficult to build generalised models of object and scene categories using SIFT keypoints which are very specific to visual attributes of particular images. However it is also possible that the SIFT-S-m signature does not preserve enough useful information from these keypoints. We try our SIFT feature on the COIL-100 dataset to see whether it performs well in an object matching setting, where one would naturally expect SIFT itself to work well. We evaluate 3 different version of our SIFT feature: signatures with 16 and 64 clusters, and a signature where each individual keypoint represets an equally weighted cluster centroid. The latter is used to establish how much information is lost through clustering keypoints in each signature. Aditionally we try the Log-Gabor-S-16 and Log-Gabor-S-64 features on the same dataset to see if object matching can be performed using simpler features. Table 3.15 shows that using 64 clusters per signature is almost the same as not clustering SIFT keypoints at all, though the latter comes at a very high computational cost when computing EMD between signatures. Mean average precision of 0.8903 can be considered very good performance for a dataset with 100 object categories. Using 16 clusters per signature results in substantial loss of information and the precision drops to 0.8078, however this is still far above random chance. Interestingly, using the Log-Gabor signature produces competitive results on this dataset. Using just 16 wavelet clusters per signature results in mean average precision of 0.8609.

The outlook is different, however, when we compare SIFT to Log-Gabor on the modified Corel dataset. Table 3.16 ranks its keywords according to the ratio of SIFT average precision to that of random retrieval. In most cases, Log-Gabor-S-16 performs substantially better than SIFT-S-16. This shows that Log Gabor wavelet feature is more robust for automated image annotation of this dataset than our SIFT feature.

Keyword	SIFT-S-16	Random	Log-Gabor-S-16
tracks	0.2765	0.0214	0.7810
polar	0.2496	0.0218	0.6269
bear	0.3302	0.0373	0.6017
plane	0.4912	0.0609	0.7176
flowers	0.3583	0.0513	0.2459
jet	0.5576	0.0870	0.7608
horses	0.2336	0.0396	0.6302
sun	0.2862	0.0486	0.3928
valley	0.0941	0.0166	0.2577
birds	0.2392	0.0432	0.2313
close-up	0.1275	0.0231	0.1474
clouds	0.3127	0.0624	0.1729
field	0.1566	0.0347	0.4537
snow	0.2311	0.0576	0.3324
cars	0.2785	0.0705	0.5552
leaf	0.1051	0.0267	0.1889
plants	0.1287	0.0401	0.1127
boats	0.1030	0.0322	0.2140
rocks	0.1719	0.0543	0.1334
stone	0.1203	0.0396	0.2480
bridge	0.1091	0.0371	0.1728
street	0.1285	0.0471	0.3133
grass	0.2678	0.0993	0.2622
sky	0.4499	0.1875	0.4409
garden	0.1030	0.0436	0.1297
house	0.0762	0.0330	0.1181
ruins	0.0530	0.0235	0.0814
buildings	0.2533	0.1192	0.3011
sand	0.1866	0.0886	0.1075
people	0.3319	0.1694	0.3657
water	0.4168	0.2217	0.4808
mountain	0.1853	0.1003	0.2022
statue	0.0398	0.0276	0.1328
tree	0.3256	0.2413	0.3162
hills	0.0541	0.0730	0.0725
beach	0.0545	0.1511	0.1472
Average	0.2191	0.0703	0.3180

Table 3.16: SIFT-S-16 and Log-Gabor-S-16 on the modified Corel dataset. Keywords for which the Log-Gabor signature feature performs better than the SIFT feature are highlighted in bold.

3.3 Practical application: refinement of Internet image search results

The ability to rank images by probability of depicting visual concepts may be of practical significance to existing Internet image search engines such as Google[7] and Yahoo[8]. As of May 2007, their image indexing method appears to be that of using collateral text data, such as image filenames or web page content. This method is sufficient for many types of queries but it suffers from two problems: relevant images that lack appropriate metadata will not be retrieved and irrelevant images with erroneous metadata may be returned. To investigate how automated image annotation can be helpful in improving Internet image search in such situations we designed a search engine called Behold (Yavlinsky, 2005). This search engine implements the nonparametric image annotation method described in this chapter (without data reduction). The search engine has been demonstrated at major international conferences (Yavlinsky et al., 2006; Heesch et al., 2006). This section uses it to demonstrate that, in addition to annotating unlabelled images automatically, text-based Internet image search results can be *improved* by re-ranking the returned images by appropriate annotation probabilities.

All sample screenshots in this section show Behold running on a database of 1.13 million images spidered from the UK, US and Canadian university websites. The search engine can be accessed online at `http://www.beholdsearch.com/phd-demo/`.

3.3.1 Training data and image features

Corel images and images downloaded from Flickr were used for creating models of 57 different keywords, which were then used to index images from the web. 14,456 images were used for training keyword annotation models.

Global colour and texture features were used to model image densities. The colour feature was similar to the MargCIE-T-3×3 described earlier with the exception that Hue, Saturation and Value colour space was used instead of CIE*Lab* one. The texture features were Tamura-T-3×3 and the standard Gabor filter bank (6 scales × 4 orientations, Howarth and Rüger 2004). Once the features were generated, annotation of the 1,131,605 images took 65 hours on a 3.0 GHz Intel 2006-vintage computer.

3.3.2 Metadata

For every image downloaded from the Internet, its URL and a copy of the webpage that contained it are kept. Keywords are extracted from both for indexing the image in the traditional manner. From the image URL we extract longest non-overlapping filename substrings that match terms in a dictionary

[7] http://images.google.com

[8] http://images.search.yahoo.com

of over 50,000 English words[9]. The image ALT-tag information in the corresponding webpage was also used. The standard Boolean matching function is applied when querying the metadata index.

3.3.3 Keyword search

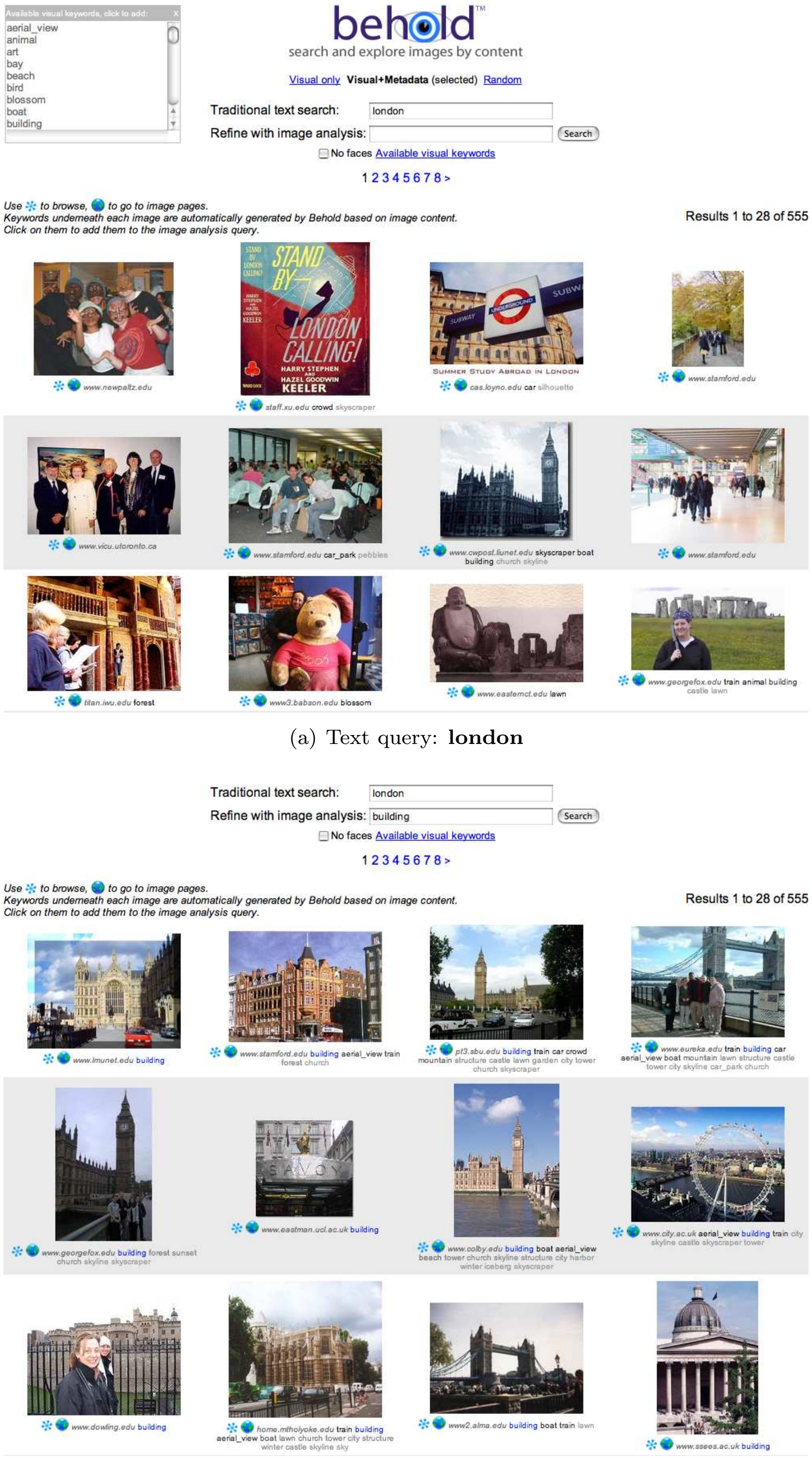
Behold's keyword search combines traditional text (metadata) queries with those based on automated image annotation. The user interface provides a separate search box for keywords intended for esach modality: **traditional text search** and **refine with image analysis**. The latter is intended to make the user aware that the keywords entered into this field will be matched against automatically assigned annotations of images. Multiple keywords can be entered into both fields.

The default search scenario is that the user queries traditional metadata to get an initial pool of results, and then refines his or her search results by specifying keywords that must be matched in the automatically generated annotations. Suppose, for example, that the user is looking for London's architectural pictures. A suitable query would consist of 'London' entered in the text search field and 'building' in the image analysis field. Behold will process this query by first retrieving images based on the presence of 'london' in image metadata, and would then *re-rank* them according to the probability of the keyword 'building' based on the visual contents of images[10]. This can improve results when the word 'London' is present in metadata of relevant images but the word 'building' is not. This situation is illustrated in Figure 3.6. When the metadata query is a single word the results can be re-ranked by the probability of the matching annotation, if the latter is present in the annotation vocabulary. This can be used to improve the quality of the results. An example of this type of refinement for the query 'beach' is shown in Figure 3.7. Even though the annotation vocabulary is limited, the combination of metadata and annotation querying is a powerful way to search. An example of refining the metadata query 'bus' with the annotation query 'car' is shown in Figure 3.8.

It is also possible to use automated image annotation search on its own within Behold. This is suitable when results of a given query using traditional text search are poor and, at the same time, desired keywords exist in the automated annotation vocabulary. For this the user must click the link above the search box to change into **Visual only** mode. As in the default mode, the system will make the user aware of keywords that are available in the vocabulary. Image metadata will be ignored during this type of search. Figure 3.9 illustrates this type of querying.

[9] http://wordlist.sourceforge.net/

[10] The idea of retrieving images based on metadata and re-ranking according to keyword probabilities has been prevously applied by many participants of NIST's TRECVID workshop. See, for example, Snoek et al. (2004); Hauptmann et al. (2004).


Available visual keywords, click to add:
aerial_view
animal
art
bay
beach
bird
blossom
boat
building
behold™
search and explore images by content
Visual only
Visual+Metadata (selected)
Random
Search
Available visual keywords
1 2 3 4 5 6 7 8 >
www.newpaltz.edu
staff.xu.edu crowd skyscraper
cas.loyno.edu car silhouette
www.stamford.edu
www.vicu.utoronto.ca
www.stamford.edu car_park pebbles
www.cwpost.liunet.edu skyscraper boat building church skyline
www.stamford.edu
titan.iwu.edu forest
www3.babson.edu blossom
www.easternct.edu lawn
www.georgefox.edu train animal building castle lawn

Traditional text search: london

Refine with image analysis:

□ No faces

Use to browse, to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 555

(a) Text query: **london**

Search
Available visual keywords
1 2 3 4 5 6 7 8 >
www.lmunet.edu building
www.stamford.edu building aerial_view train forest church
pt3.sbu.edu building train car crowd mountain structure castle lawn garden city tower church skyscraper
www.eureka.edu train building car aerial_view boat mountain lawn structure castle tower city skyline car_park church
www.georgefox.edu building forest sunset church skyline skyscraper
www.eastman.ucl.ac.uk building
www.colby.edu building boat aerial_view beach tower church skyline structure city harbor winter iceberg skyscraper
www.city.ac.uk aerial_view building train city skyline castle skyscraper tower
www.dowling.edu building
home.mtholyoke.edu train building aerial_view boat lawn church tower city structure winter castle skyline sky
www2.alma.edu building boat train lawn
www.ssees.ac.uk building

Traditional text search: london

Refine with image analysis: building

□ No faces

Use to browse, to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 555

(b) Text query: **london**, visual analysis query: **building**

Figure 3.6: Refining a metadata query for London through automated annotation search. Searching for 'london building' using just the text search returns one result.

3.3.4 Performance evaluation

We tried to measure the effect of re-ranking by annotation probability that we illustrated in Figure 3.7. Unfortunately it is prohibitive to carry out relevance judgements for the entire 1.13 million images, yet without these judgements a thorough automated performance evaluation is not possible. We used precision at 20 and precision at 100 as measures of retrieval accuracy on 11 one-word queries: 'building', 'church', 'castle', 'flower', 'garden', 'boat', 'beach', 'grass', 'mountain', 'sunset' and 'car'. Relevant images were marked manually for every query result. Images were judged relevant if they were photographic and clearly depicted the query concept, or if they were a high-detail or a photo-realistic illustration of that concept.

Table 3.17 shows the number of relevant images returned by Behold in the top 20 results. Three different querying modes were used: visual (automated annotation only), text (metadata) and the visual+text combination, where the metadata results are re-ranked by the corresponding annotation probability. A pool of 20 randomly selected images was used each time to check how often images relevant to the query occur by chance. Table 3.18 shows results for the same queries in the top 100 results and Table 3.19 shows precision averages. From these three tables one can see that while searching the metadata index is superior to using automated annotation on its own, the combination of the two results in impressive 62% increase over metadata search performance. This is the case for both precision at 20 and precision at 100. Improvement is particularly noticeable for the 'car' query. This can be explained by the poor quality of the metadata index for this term: car is a very short string and may commonly occur as a substring within words not contained in our dictionary. This will result in many false positives, and the re-ranking by annotation probability is most helpful here.

To get an indication of our metadata indexing quality we tried the same 11 queries in Google Image Search. We applied the same relevance judgement procedure as before, and the results are presented in Table 3.20. Google's retrieval precision is above Behold's text-only search, however, we cannot use these results for direct comparison. Google has to index at least 1000 times more images, which come from a much broader range of sources than the university websites we used. Unlike Behold, Google associates web page content with images and is likely to use a different retrieval criterion than the Boolean matching function used by the former. Therefore, image indexing is a much greater challenge for Google than it is for Behold. Nonetheless, one can see that the precision figures are in the same range for both search engines. This suggests that refinement of search results through automated annotation might be a promising direction for larger search engines like Google as well.

Query	Visual	Text	Visual+Text	Random
Building	20	13	19	4
Church	5	12	20	0
Castle	5	13	19	0
Flower	11	17	20	0
Garden	3	11	19	0
Boat	9	13	19	0
Beach	8	13	20	0
Grass	15	12	19	3
Mountain	11	9	20	0
Sunset	20	17	20	0
Car	7	1	18	0

Table 3.17: Number of relevant images returned by Behold in the top 20 results

Query	Visual	Text	Visual+Text	Random
Building	96	69	96	9
Church	20	55	84	0
Castle	19	48	88	0
Flower	40	73	96	1
Garden	12	44	89	0
Boat	36	42	75	0
Beach	24	50	99	0
Grass	83	47	71	9
Mountain	33	55	72	0
Sunset	87	84	100	0
Car	43	12	69	2

Table 3.18: Number of relevant images returned by Behold in the top 100 results

Mode	Prec. @ 20	Prec @ 100
Visual	0.5182	0.4482
Text	0.5955	0.5264
Visual+Text	0.9682	0.8536
Random	0.0318	0.0191

Table 3.19: Precision average for 11 queries in Behold

Query	# rel. in top 20	# rel. in top 100
Building	18	61
Church	14	76
Castle	19	73
Flower	10	54
Garden	19	57
Boat	15	73
Beach	17	62
Grass	15	59
Mountain	11	32
Sunset	15	70
Car	16	51
Precision	0.7682	0.6073

Table 3.20: Precision of the 11 queries in Google Image Search

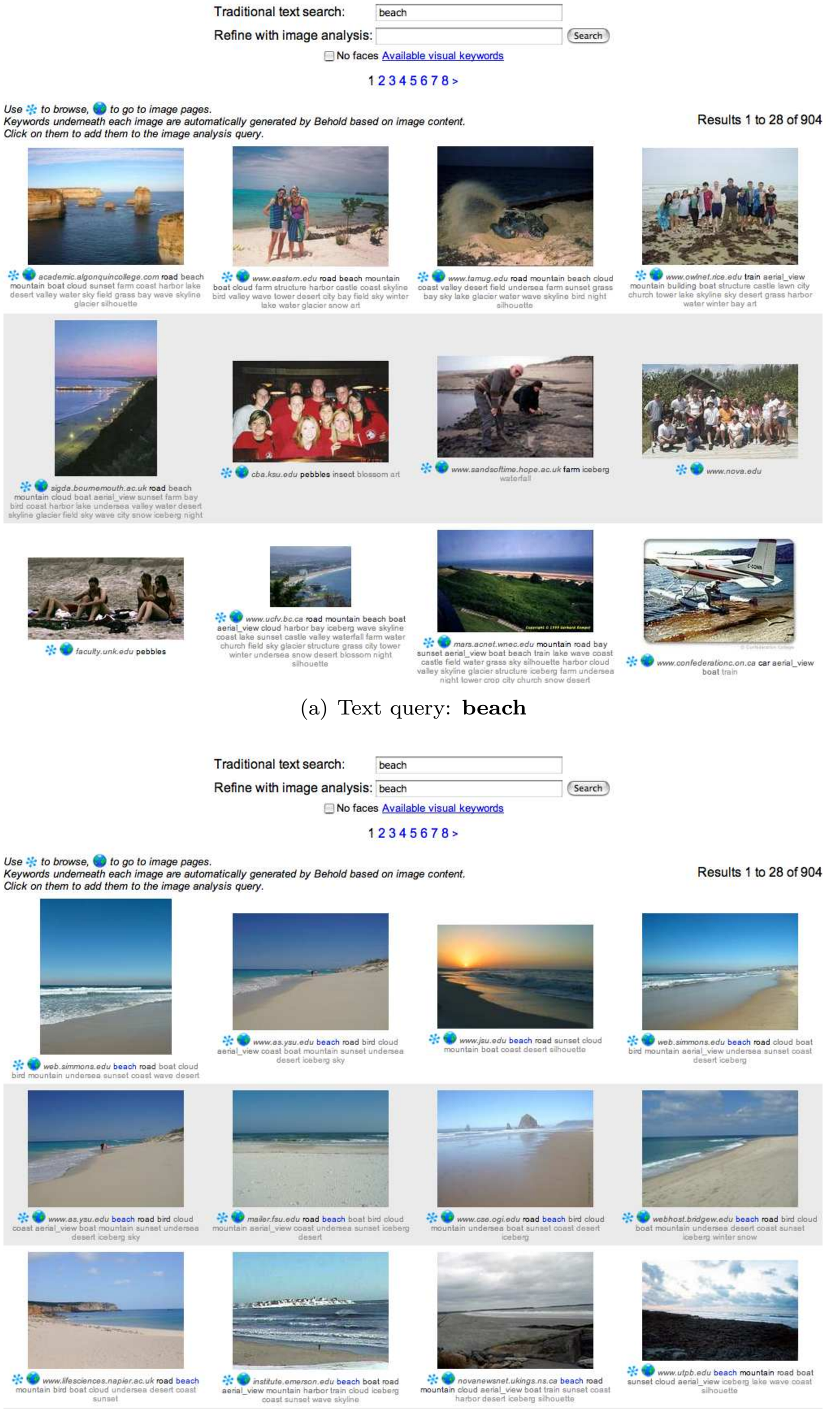
3.4 Conclusions

We have presented a simple framework for automated image annotation based on nonparametric density estimation. Consistently with earlier findings by Oliva and Schyns (2000) and Torralba and Oliva (2003), we have shown that under this framework very simple global image properties can yield reasonable annotation accuracies. In particular, we find that using merely colour information can achieve 'state of the art' performance for the Corel dataset. Suitable combinations of global colour and texture information result in retrieval accuracy that rivals the best benchmark annotation results. Such combinations are also suitable for modelling keyword distributions in the two new datasets we have introduced in this chapter.

We have proposed a novel density estimation technique which relies on the Earth Mover's Distance metric and have shown it to be more effective than the standard way of modelling probability density functions of histograms vectors. Furthermore, we have shown that it is possible to accelerate EMD density estimation using simple training data reduction techniques. Under this density estimation approach, the Log-Gabor wavelet feature appears to perform better than our more sophisticated SIFT-based image representation. We shall discuss the possible reasons behind this in Section 6.3.2.

Finally, we have shown how automated image annotation can be used to improve text-based Internet image search on a database of over 1.13 million images downloaded from the World Wide Web.

Having found that simple image features are suitable for automated image annotation, we move on to exploring how characteristic feature values vary across different image collections. In the next chapter we shall see how performance is affected when keyword models estimated on one collection are used to annotate images from another. We shall also see whether low-level feature distributions are similar for similar keywords in different datasets, and whether it is possible to predict annotation accuracy of a keyword based on its distribution.



Traditional text search: beach

Refine with image analysis: Search

□ No faces Available visual keywords

1 2 3 4 5 6 7 8 >

Use ❄ to browse, 🌍 to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 904

❄ 🌍 *academic.algonquincollege.com* road beach mountain boat cloud sunset farm coast harbor lake desert valley water sky field grass bay wave skyline glacier silhouette

❄ 🌍 *www.eastern.edu* road beach mountain boat cloud farm structure harbor castle coast skyline bird valley wave tower desert city bay field sky winter lake water glacier snow art

❄ 🌍 *www.tamug.edu* road mountain beach cloud coast valley desert field undersea farm sunset grass bay sky lake glacier water wave skyline bird night silhouette

❄ 🌍 *www.owlnet.rice.edu* train aerial_view mountain building boat structure castle lawn city church tower lake skyline sky desert grass harbor water winter bay art

❄ 🌍 *sigda.bournemouth.ac.uk* road beach mountain cloud boat aerial_view sunset farm bay bird coast harbor lake undersea valley water desert skyline glacier field sky wave city snow iceberg night

❄ 🌍 *cba.ksu.edu* pebbles insect blossom art

❄ 🌍 *www.sandsoftime.hope.ac.uk* farm iceberg waterfall

❄ 🌍 *www.nova.edu*

❄ 🌍 *faculty.unk.edu* pebbles

❄ 🌍 *www.ucfv.bc.ca* road mountain beach boat aerial_view cloud harbor bay iceberg wave skyline coast lake sunset castle valley waterfall farm water church field sky glacier structure grass city tower winter undersea snow desert blossom night silhouette

❄ 🌍 *mars.acnet.wnec.edu* mountain road bay sunset aerial_view boat beach train lake wave coast castle field water grass sky silhouette harbor cloud valley skyline glacier structure iceberg farm undersea night tower crop city church snow desert

❄ 🌍 *www.confederationc.on.ca* car aerial_view boat train

(a) Text query: **beach**

Traditional text search: beach

Refine with image analysis: beach Search

□ No faces Available visual keywords

1 2 3 4 5 6 7 8 >

Use ❄ to browse, 🌍 to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 904

❄ 🌍 *web.simmons.edu* beach road boat cloud bird mountain undersea sunset coast wave desert

❄ 🌍 *www.as.ysu.edu* beach road bird cloud aerial_view coast boat mountain sunset undersea desert iceberg sky

❄ 🌍 *www.jsu.edu* beach road sunset cloud mountain boat coast desert silhouette

❄ 🌍 *web.simmons.edu* beach road cloud boat bird mountain aerial_view undersea sunset coast desert iceberg

❄ 🌍 *www.as.ysu.edu* beach road bird cloud coast aerial_view boat mountain sunset undersea desert iceberg sky

❄ 🌍 *mailer.fsu.edu* road beach boat bird cloud mountain aerial_view coast undersea sunset iceberg desert

❄ 🌍 *www.cse.ogi.edu* road beach bird cloud mountain undersea boat sunset coast desert iceberg

❄ 🌍 *webhost.bridgew.edu* beach road bird cloud boat mountain undersea desert coast sunset iceberg winter snow

❄ 🌍 *www.lifesciences.napier.ac.uk* road beach mountain bird boat cloud undersea desert coast sunset

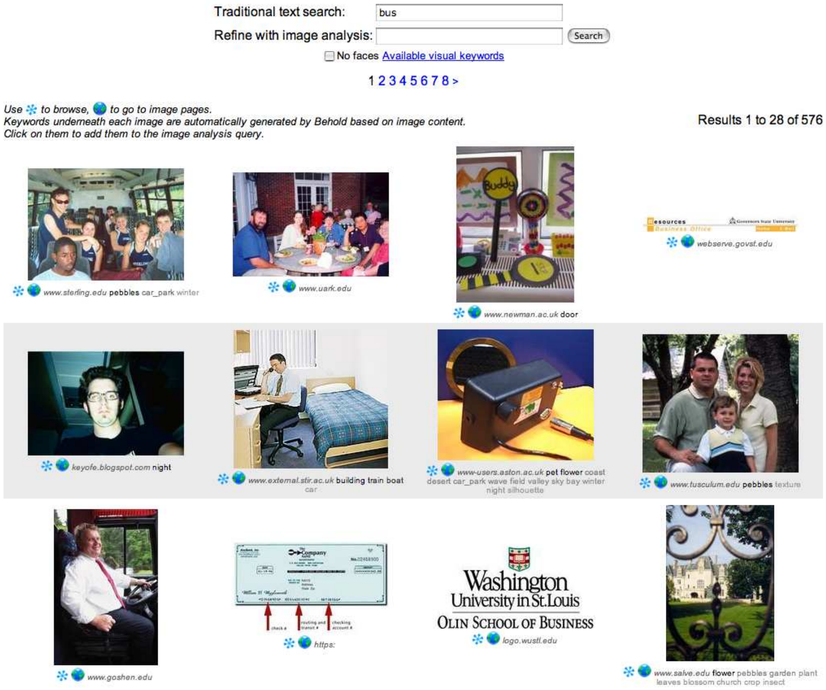
❄ 🌍 *institute.emerson.edu* beach boat road aerial_view mountain harbor train cloud iceberg coast sunset wave skyline

❄ 🌍 *novanewsnet.ukings.ns.ca* beach road mountain cloud aerial_view boat train sunset coast harbor desert iceberg silhouette

❄ 🌍 *www.utpb.edu* beach mountain road boat sunset cloud aerial_view iceberg lake wave coast silhouette

(b) Text query: **beach**, visual analysis query: **beach**

Figure 3.7: Refining metadata search for beach.



Traditional text search: bus

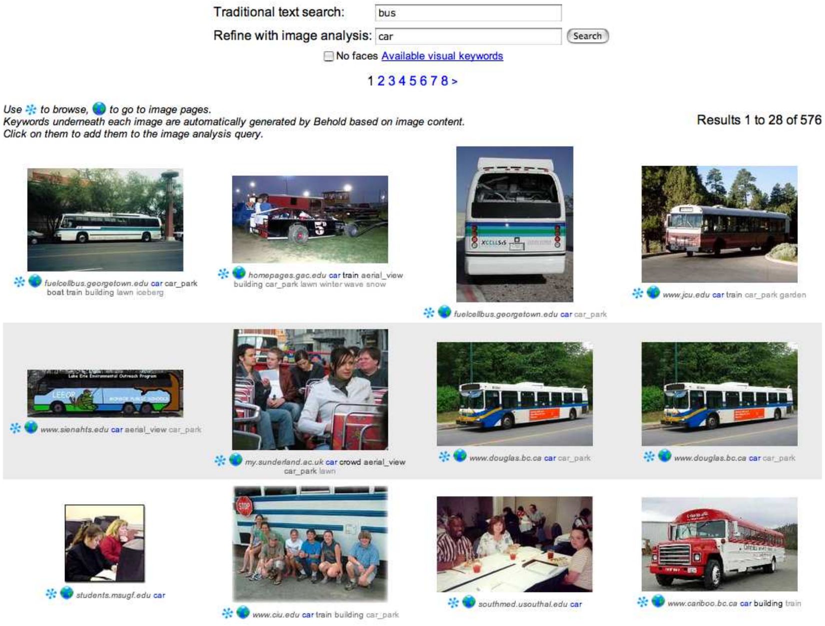
Refine with image analysis:

Use ❄ to browse, 🌐 to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 576

Search
□ No faces Available visual keywords
1 2 3 4 5 6 7 8 >
www.sterling.edu pebbles car_park winter
www.uark.edu
www.newman.ac.uk door
webserve.govst.edu
keyofe.blogspot.com night
www.external.stir.ac.uk building train boat car
www-users.aston.ac.uk pet flower coast desert car_park wave field valley sky bay winter night silhouette
www.tusculum.edu pebbles texture
www.goshen.edu
https:
Washington University in St. Louis
Olin School of Business
logo.wustl.edu
www.salve.edu flower pebbles garden plant leaves blossom church crop insect

(a) Text query: **bus**



Traditional text search: bus

Refine with image analysis: car

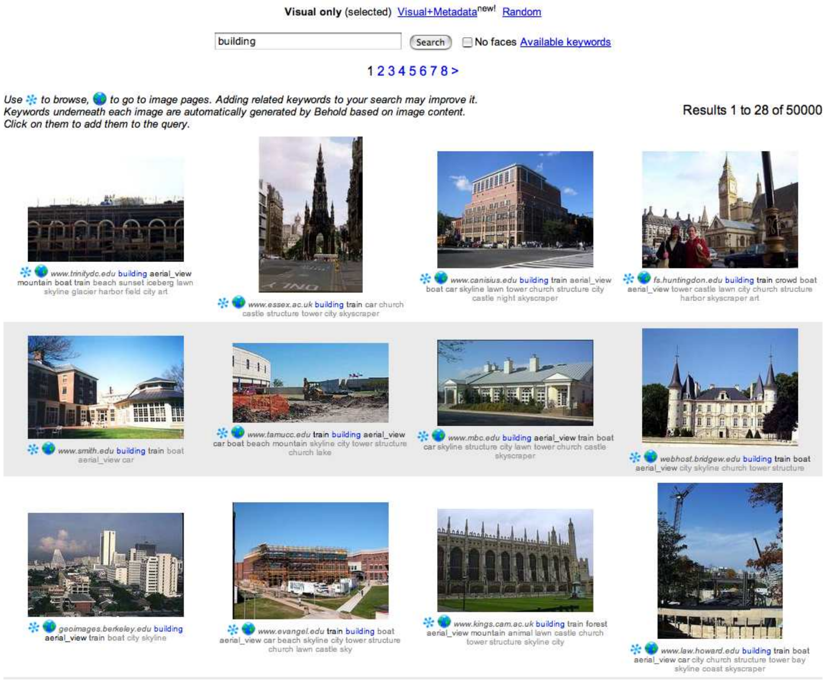
Use ❄ to browse, 🌐 to go to image pages.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the image analysis query.

Results 1 to 28 of 576

Search
□ No faces Available visual keywords
1 2 3 4 5 6 7 8 >
fuelcellbus.georgetown.edu car car_park boat train building lawn iceberg
homepages.gac.edu car train aerial_view building car_park lawn winter wave snow
fuelcellbus.georgetown.edu car car_park
www.jcu.edu car train car_park garden
www.sienahts.edu car aerial_view car_park
my.sunderland.ac.uk car crowd aerial_view car_park lawn
www.douglas.bc.ca car car_park
www.douglas.bc.ca car car_park
students.msugf.edu car
www.ciu.edu car train building car_park
southmed.usouthal.edu car
www.cariboo.bc.ca car building train

(b) Text query: **bus**, visual analysis query: **car**

Figure 3.8: Refining metadata search for bus.



Visual only (selected) Visual+Metadata$^{new!}$ Random
building
Search
□ No faces Available keywords
1 2 3 4 5 6 7 8 >

Use ❄ to browse, 🌍 to go to image pages. Adding related keywords to your search may improve it.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the query.

Results 1 to 28 of 50000

www.trinitydc.edu building aerial_view mountain boat train beach sunset iceberg lawn skyline glacier harbor field city art

www.essex.ac.uk building train car church castle structure tower city skyscraper

www.canisius.edu building train aerial_view boat car skyline lawn tower church structure city castle night skyscraper

fs.huntingdon.edu building train crowd boat aerial_view tower castle lawn city church structure harbor skyscraper art

www.smith.edu building train boat aerial_view car

www.tamucc.edu train building aerial_view car boat beach mountain skyline city tower structure church lake

www.mbc.edu building aerial_view train boat car skyline structure city lawn tower church castle skyscraper

webhost.bridgew.edu building train boat aerial_view city skyline church tower structure

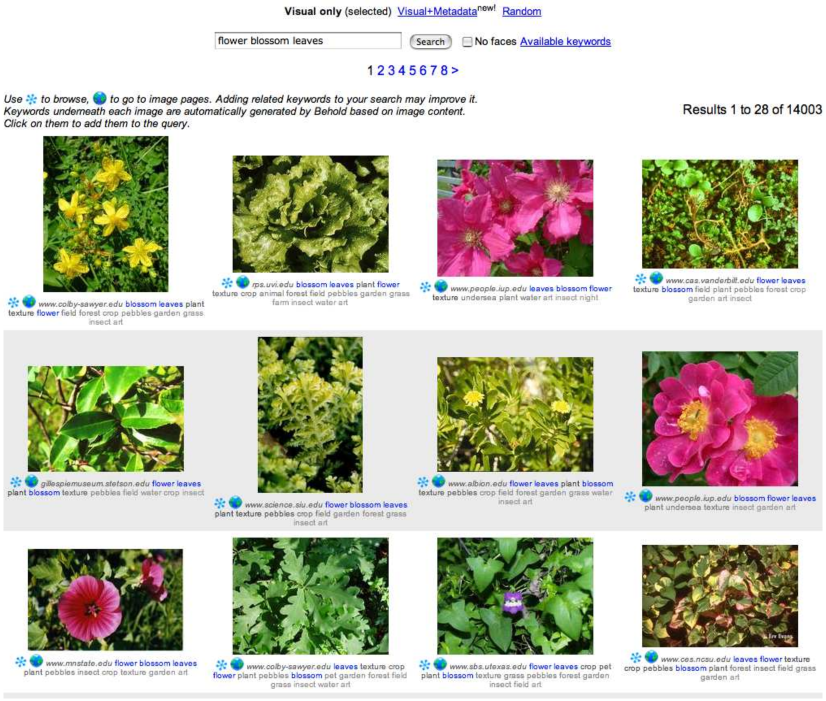
geoimages.berkeley.edu building aerial_view train boat city skyline

www.evangel.edu train building boat aerial_view car beach skyline city tower structure church lawn castle sky

www.kings.cam.ac.uk building train forest aerial_view mountain animal lawn castle church tower structure skyline city

www.law.howard.edu building train boat aerial_view car city church structure tower bay skyline coast skyscraper

(a) Visual analysis query: **building**



Visual only (selected) Visual+Metadata$^{new!}$ Random
flower blossom leaves
Search
□ No faces Available keywords
1 2 3 4 5 6 7 8 >

Use ❄ to browse, 🌍 to go to image pages. Adding related keywords to your search may improve it.
Keywords underneath each image are automatically generated by Behold based on image content.
Click on them to add them to the query.

Results 1 to 28 of 14003

www.colby-sawyer.edu blossom leaves plant texture flower field forest crop pebbles garden grass insect art

rps.uvi.edu blossom leaves plant flower texture crop animal forest field pebbles garden grass farm insect water art

www.people.iup.edu leaves blossom flower texture undersea plant water art insect night

www.cas.vanderbilt.edu flower leaves texture blossom field plant pebbles forest crop garden art insect

gillespiemuseum.stetson.edu flower leaves plant blossom texture pebbles field water crop insect

www.science.siu.edu flower blossom leaves plant texture pebbles crop field garden forest grass insect art

www.albion.edu flower leaves plant blossom texture pebbles crop field forest garden grass water insect art

www.people.iup.edu blossom flower leaves plant undersea texture insect garden art

www.mnstate.edu flower blossom leaves plant pebbles insect crop texture garden art

www.colby-sawyer.edu leaves texture crop flower plant pebbles blossom pet garden forest field grass insect water art

www.sbs.utexas.edu flower leaves crop pet plant blossom texture grass pebbles forest garden insect field art

www.ces.ncsu.edu leaves flower texture crop pebbles blossom plant forest insect field grass garden art

(b) Visual analysis query: **flower blossom leaves**

Figure 3.9: Querying using only automated annotation

Chapter 4

Evaluation of Global Feature Reliability

In the previous chapter we have outlined our framework for automated image annotation using simple visual features. We have shown that it is capable of outperforming state-of-the-art retrieval results on one standard dataset and that it facilitates reasonable retrieval accuracy on two large, realistic image collections. However, the accuracy of a feature extraction technique on a small number of different datasets is typically insufficient for assessing its generality. Unrealistically high retrieval precision could be obtained because our global features pick up on image artefacts that are only useful within a particular dataset. A thorough evaluation on a significantly larger scale is difficult for a number of reasons: manually labelling images is a slow and tedious process, labels assigned to images by humans are inherently subjective and different labels will often refer to the same underlying concept. In this chapter we suggest a number of additional measures for evaluating the generality of keyword models in visual feature space, briefly described below.

If our simple features capture real-world, visual patterns relevant to the keywords that we model, we would intuitively expect that

- keywords relating to semantically similar visual concepts in the same dataset will have similar density functions in the low level feature space
- keywords relating to identical concepts in *different* datasets will likewise have similar density functions
- automatically annotating images from one dataset using keyword models estimated on another would give reasonable retrieval precision

Admittedly, ‘semantic similarity’ is a subjective term. In our case we are interested in similarities between keywords such as *field* and *grass* or *person* and *face*, where both keywords pertain to the same object or scene. It makes particular sense to compare feature distributions of similar keywords in different datasets. One way to view a dataset is as a non-random sample of pictures drawn from the universe of all possible natural images. The non-random bias is introduced by the dataset's creator. The person compiling the image collection might include only a certain type of images depicting the given concept, owing to a personal preference. When using the same dataset for both training and evaluation, this homogeneity may result in an increase of retrieval accuracy for that concept. This phenomenon has been previously observed in a content-based image retrieval setting by Müller et al. (2002). This bias may affect different datasets to different degrees.

In our case we would like to distinguish real contextual dependencies in images from the above bias. For example, suppose that the images of horses all contain green grass, the latter being much easier to detect using a global feature rather than a horse itself. Is such contextual dependence general or is it biased towards the given dataset? By considering keyword model similarities across different datasets one would get an indication of how reliable such contextual cues are. In this chapter we use the Earth Mover's Distance (Rubner, 1998) to quantify similarities between keyword feature distributions.

Some keywords may not correlate to our image features, in which case it would be good to remove such keywords from the vocabulary prior to automated annotation. Typically, such keywords will relate to complex objects appearing in many different contexts (such as pictures of people) or abstract concepts (e.g. 'Art Deco buildings'). In this chapter we propose a simple measure of visualness of a keyword and investigate its correlation to the keyword's retrieval precision.

4.1 Related work

4.1.1 Evaluation of feature generality

At the time of writing there appear to be no publications that contain comparisons between low-level feature distributions of identical concepts in different datasets, or that of similar concepts within the same dataset. However, research by Müller et al. (2002) highlights the degree to which artificial homogeneity of images for a particular concept may unrealistically increase retrieval precision in a content-based image retrieval setting. They report that for the Corel dataset, the removal of visually dissimilar images from a particular category and elimination of poorly performing image classes significantly improves retrieval precision. Reflecting on these results one may wonder whether current low-level feature extraction techniques would work at all in the real world (e.g. on over a million images from the World Wide Web). If modifications of a finite evaluation dataset result in such variations in performance, are our experimental results significant?

Similar problems have been encountered in the object recognition domain. Ponce et al. (2006) point out that restrictions in object-recognition datasets with respect to object viewpoints, orientations, sizes, positions and backgrounds, allow rather simple algorithms to recognise object classes well. When these restrictions are lifted, however, recognition accuracy drops drastically. Of course, from our point of view, such restrictions could be interpreted as contextual regularities which we strive to exploit. However, as noted in the previous section, it is important to assess the generality of these detected regularities.

4.1.2 Visualness measures

Rao et al. (2002) propose a method of measuring the 'complexity' of an image database. They extend the information-theoretic measure of text corpus perplexity to the image domain. Since perplexity analyses the entropy of word sequences, which are discrete tokens, image features must be transformed into a similar representation for this measure to be applicable. The authors solve this problem by quantising image blocks into a codebook of feature vectors. Images are then described as 2-dimensional code sequences. They define the complexity of an image collection as a function of the entropy of code sequences in the collection. The authors report correlation between the complexity measure of a dataset and the accuracy of content-based image retrieval on that dataset. One possible drawback of this approach is that information is necessarily lost through vector quantisation.

Yanai and Barnard (2005) look at the problem in a different way. Instead of measuring the complexity of the entire image database, they focus on the visualness of individual annotation keywords. In their framework each image is partitioned into a number of segments using a standard image segmentation algorithm. They assume that training images are labelled manually with multiple keywords, but that correspondence between image regions and keywords is not provided. The initial stage of their approach estimates this correspondence automatically using an EM-like iterative algorithm. Once regions are thus associated with the given keyword, entropy of these regions' visual features is computed. Lower entropy indicates greater visualness of the keyword. The authors order adjective keywords according to this measure and they report that the resulting keyword visualness ranking is intuitive. A possible problem with this approach is the need to automatically estimate correspondence between image segments and keywords, which may not always yield accurate results and thus introduce an error into the final entropy calculation. An inaccurate image segmentation algorithm may further compound this problem.

In this chapter we propose a way to measure visualness that avoids both feature quantisation and the need for image segmentation.

4.2 Image data

In this chapter we evaluate the proposed techniques on seven different datasets. Five of them have been introduced in the previous chapter: Corel, modified Corel, Getty, modified Getty and Web images. As described before, the modified Getty and Corel datasets consist of the same images as their unmodified counterparts but the vocabularies have been changed. We also introduce two additional datasets, described below.

Corel-16k. This collection consists of 16,215 images from the Corel 380,000 image collection. We compiled a diverse vocabulary of 203 keywords such that it would maximally overlap with the vocabulary of the Getty collection. The collection was split into 8,058 training images and 8,157 test images. The training set was further split into 4,000 training 4,058 validation images for optimising the kernel bandwidth. When annotating the test part of this collection with keyword models estimated on the training part using the MargCIE-T-3×3+Tamura-T-3×3+Log-Gabor feature combination the mean average precision of single word queries is 0.2993.

Flickr. This collection was obtained from the Flickr[1] photo-sharing website. It consists of 32,004 images downloaded from the Flickr group "JPEG Magazine" – a group dedicated to unaltered images taken by Flickr users. Only images shared under the 'Creative Commons' license were downloaded. User generated 'tags' were the only readily available annotations. Given that these tags have not been validated by professional image collection curators, they are not used as a basis for formal retrieval evaluation. Instead they are only used for visual verification of the visualness measure on real world image data. From the 500 most frequently used tags we selected 173 that we thought were not unreasonable to attempt to model using low-level visual features. Below is a list of tags randomly sampled from this selection for indication purposes:

horse, buildings, female, square, mountains, rocks, animal, garden, brown, city, rural, green, rain, fire, pink, bike, building, architecture, dog, night, stairs, light, snow, gold, boy, tower, women, farm, pier, skyline, stone, girl, streetart, window. Full details of this dataset can be found online[2].

4.3 Comparing feature distributions of similar keywords

In this section we consider the similarity between keyword distributions in our chosen feature space. If semantically similar keywords have similar respective distributions, there is a stronger chance that our feature representation is generic. We use the Earth Mover's Distance metric, discussed in Chapter 3, as a measure of keyword distribution similarity. A simple k-means algorithm is applied to each keyword sample of feature vectors T_w to generate its distribution signature, where k is set to 100. If the sample contains 100 or fewer vectors each vector becomes a cluster centroid with an equal weight assigned to it. For comparing keywords we use the MargCIE-T-3×3, Tamura-T-3×3 and Log-Gabor features individually, and the joint feature in which the outputs of the above three features are concatenated

[1] http://www.flickr.com

[2] http://www.beholdsearch.com/research/datasets/

into a single vector – MargCIE-T-3×3+Tamura-T-3×3+Log-Gabor. All feature vector components are normalised by their standard deviation on a withheld sample of images.

4.3.1 Keyword feature distribution similarity within datasets

First we analyse keyword distribution similarities within individual datasets. For an image collection A we define a vocabulary S_A which is a subset the collection's full vocabulary W_A. For each keyword s in S_A we calculate the EMD values between its low-level feature distribution D_s and that of every keyword w in W_A, D_w. We rank keywords w according to the proximity of their distributions to D_s.

Table 4.4 shows such rankings for a few representative keywords from the Getty dataset for two features: the joint feature and Tamura-T-3×3. S_{Getty} is the vocabulary of the modified Getty dataset (36 keywords in $W_{\text{modified Getty}}$) and W_{Getty} is the 247 keywords used in the original Getty collection. Tables A.1-A.3 in Appendix A show rankings for all keywords in S_{Getty} for all four features. A table entry consists of a keyword s followed by lists of most similar keywords – each row shows keyword similarity under each feature. The order of these four rows corresponds to the order of the features listed above. Each list shows 9 most similar keywords for the particular feature, ordered by the proximity of their distributions to D_s. Words which the author judged as semantically similar to each keyword being considered are emphasised in italics. Although these judgements are subjective (thereby preventing us from carrying out a formal quantitative evaluation) they help to show that, under all chosen features, keyword distribution similarities within this dataset are often semantically meaningful, as we would hope.

We repeated this experiment on the Corel-16k dataset. $S_{\text{Corel-16k}}$ consists 32 keywords similar to the ones in S_{Getty} and are compared each against $W_{\text{Corel-16k}}$, the full 203-word vocabulary. Results for selected keywords under the joint feature and Tamura-T-3×3 are presented in Table 4.5. Tables A.4-A.6 in Appendix A show rankings for all keywords in S_{Getty} for all four features. One can observe that for this dataset the inter-keyword similarities are also quite often meaningful. However in both datasets some of the closest keywords are semantically irrelevant. This underscores the limited accuracy of our chosen low-level feature representation.

Another way to visualise similarities between keyword distributions is by applying metric Multidimensional Scaling to the pairwise keyword distribution distances. Metric Multidimensional Scaling (MDS), also known as Principal Coordinates Analysis, takes a complete set of inter-point distances and creates a configuration of points in real vector space (Kruskal and Wish, 1978). The Euclidean distances between them approximately reproduce the original inter-point distances. We used the MATLAB implementation of MDS to construct 'geometric configurations' of keywords in 2 dimensions based on their inter-distribution EMD values. Figure 4.3 shows such configurations of keyword distributions from the modified Corel dataset, and 4.4 shows the same for the keywords in the modified Getty collection. Each figure shows results of MDS within the joint and Tamura feature spaces. For both datasets, the joint feature results in intuitive organisation of keywords. In the modified Getty plot for this feature, urban scene keywords *Window*, *Skyscraper*, *Urban Scene*, *Cityscape* and *Skyline* are grouped at the top of the plot, while other natural scene-based keywords are mostly located in the lower part. One can see that in modified Corel the urban-scene keywords *street*, *bridge*, *tracks*, *cars*, *buildings* and *statue* are likewise grouped at the top of the plot, separately from keywords relating to natural scenes. These intuitive keyword groupings support our hypothesis that our global features extract patterns that are meaningful in a more general sense. However, the MDS keyword plots with respect to the Tamura feature result in counter-intuitive visualisations in both cases. The latter may indicate that Tamura-T-3×3 on its own is not adequate for modelling the keywords of these two datasets. In the following section we shall see additional evidence supporting this judgement.

4.3.2 Keyword feature distribution similarity in different datasets

Next we look at keyword similarity distributions *across* datasets. For an image collection A we define a vocabulary S_A which is a subset the collection's full vocabulary W_A. Consider a different collection B with its respective vocabulary W_B. For each keyword s in S_A we calculate the EMD values between its low-level feature distribution D_s and that of every keyword w in W_B, D_w. We likewise rank keywords w according to the proximity of their distributions to D_s.

The following A/B collection pairs were evaluated: Web images/Getty, Web images/Corel-16k and Corel-16k/Getty. Vocabularies of Web images, Getty and Corel-16k share many keywords, which makes it possible to find the corresponding keywords in W_B for most keywords in S_A. We use this to count the number of keywords in S_A for which the corresponding D_w comes in the top nine closest distributions to D_s under EMD. This allows us to measure the relative generalisation ability of different feature representations more formally.

Table 4.6 shows keyword similarities for the Web images/Corel-16k dataset combination. Table 4.7 shows a subset of keyword similarities for the Corel-16k/Getty combination. In the case of Corel-16k, $S_{\text{Corel-16k}}$ consists of the same 32 keywords as in the previous experiment and $S_{\text{Web images}}$ is the same as the collections full vocabulary $W_{\text{Web images}}$. In each table, as before, 9 most similar keyword distributions are ordered by similarity, for each keyword, with respect to the joint feature and Tamura-T-3×3. Corresponding keywords are highlighted in bold. Table 4.1 summarises the number of times a corresponding keyword is found in the top 9 keywords for each feature. For Web images/Corel-16k combination there are 9 keywords that match (there are no matches in $W_{\text{Corel-16k}}$ for *Jug* and *Crowd*). For the Web images/Corel-16k 9 keywords match (no matches in W_{Getty} for *Aerial view*, *Jug* and *Crowd*). Each of the 32 keywords in $S_{\text{Corel-16k}}$ has a matching keyword in W_{Getty}. In general, the number of keyword matches significantly exceeds what one would expect to obtain at random. The joint feature appears to perform best for the Web images/Corel-16k and Corel-16k/Getty although the MargCIE-T-

	Joint feature	MargCIE-T-3×3	Tamura-T-3×3	Log-Gabor	Random
Web images/Corel-16k	6/9 (67%)	4/9 (44%)	3/9 (33%)	6/9 (66%)	9/203 (5%)
Web images/Getty	7/8 (86%)	8/8 (100%)	2/8 (25%)	6/8 (75%)	9/247 (4%)
Corel-16k/Getty	25/32 (78%)	24/32 (75%)	5/32 (16%)	9/32 (28%)	9/247 (4%)

Table 4.1: Number of times the corresponding keyword appears in the 9 most similar keyword distributions.

3×3 does slightly better for the Web images/Getty combination. Tamura-T-3×3 performs badly for all three combinations. The results for the Corel-16k / Getty combination are most significant given the number of keywords evaluated.

It is important to note that whereas previously good keyword matching was achieved for *all* features *within* datasets, this is not the case when comparing keyword distributions *across* datasets. This can be explained by a special kind of 'over-fitting'. Many semantically similar keywords often share identical images (by way of co-occurrence) and this may positively affect keyword distribution comparison in the same-dataset scenario. Tamura-T-3×3 feature performs particularly badly at the cross-dataset task, perhaps owing to its relative simplicity (3 texture properties for 9 image tiles). Recall that using this feature for generating MDS keyword plots resulted in especially geometrically-counterintuitive keyword visualisations. Together, these two results may indicate that Tamura-T-3×3 is inappropriate for image annotation on its own, despite good results reported for this feature on the Corel dataset in the previous chapter.

It is also interesting to note that the most similar keywords that do not match the target keyword exactly are also often semantically meaningful. For example, the Web images *Building exterior* keyword's nine closet keywords in Corel-16k under the joint feature are *architecture*, *building*, *buildings*, *ruins*, *structure*, *rocks*, *exterior*, *castle*, *stone* In Getty they are *House*, *Building Exterior*, *Urban Scene*, *Cityscape*, *City*, *Structure*, *Town*, *Tree*, *Window*. Such examples further bolster our hypothesis that the contextual dependencies that our features extract do indeed generalise across different image collections.

4.4 A measure of keyword visualness

Inspired by the work of Yanai and Barnard (2005) we propose our own measure of visualness of a keyword within our global feature framework. We can describe the hypothesis on which our measure is based as follows. If our feature representation is capable of reliably encoding recurring image patterns that are relevant to a particular concept, we would expect the feature values to vary little with respect to these patterns. In other words, we would expect the images of the concept to be grouped into one or more tight clusters in this feature space. Ideally, the number of such clusters would depend on the number of distinct characteristic patterns that are related to this concept. This reasoning echoes that

n	Modified Corel	Corel	Corel-16k	Getty
1%	-0.7025	-0.5340	-0.2235	-0.3468
2%	-0.6991	-0.5232	-0.1720	-0.2779
5%	-0.6935	-0.4843	-0.0985	-0.2138
10%	-0.6759	-0.4761	-0.0684	-0.2108

Table 4.2: Correlation coefficients of $\tilde{t}$ and average precision for the joint feature.

of other researchers behind relating low-level feature entropy to visualness of image samples. The lower the entropy of the features, the lower the element of surprise in the data and thus the more confident we are in what our features are measuring (the reader is referred to the textbook of Cover and Thomas (1991) for a thorough information-theoretic overview of entropy). Our visualness measure is based on the same intuition, however it avoids explicit calculation of entropy of the data sample. This is so because the latter is difficult to compute exactly for a nonparametric probability density estimate and require expensive numerical estimation techniques that rely on sampling (Viola et al., 1996). On the other hand, as stated in the previous chapter, our keyword samples are unlikely to be distributed according to simple parametric forms for which this computation would be less challenging.

A natural way to measure the relative clustering of feature vectors in a given sample is to consider the probability density at each sample point. If the probability density at most sample points is high the sample is more likely to be comprised of tight clusters. We assume that the probability density at each point is approximately inversely proportional to the average distance from the point to its n nearest neighbours. A similar assumption is made in Mitra et al. (2002), where the density at a point is assumed to be inversely proportional to that point's distance to its n^{th} nearest neighbour. The visualness measure is thus calculated as follows:

- For each point $x^{(i)}$ in a keyword sample T_w, calculate the average $L1$ distance to its n nearest neighbours in the sample. Denote this value $t^{(i)}$.
- For each point $y^{(j)}$ in a large set of randomly sampled images U repeat the above step. Denote the average nearest neighbour distance to $y^{(j)}$ as $u^{(j)}$.
- Define the relative clustering of T_w as the ratio of medians of $\tilde{t}$ and t and $\tilde{u}$ of u, $c_w = \frac{\tilde{t}}{\tilde{u}}$.
- Rank each keyword w according to its clustering value c_w. More visual keywords will have associated image samples that are more clustered and will therefore have lower values of c_w.

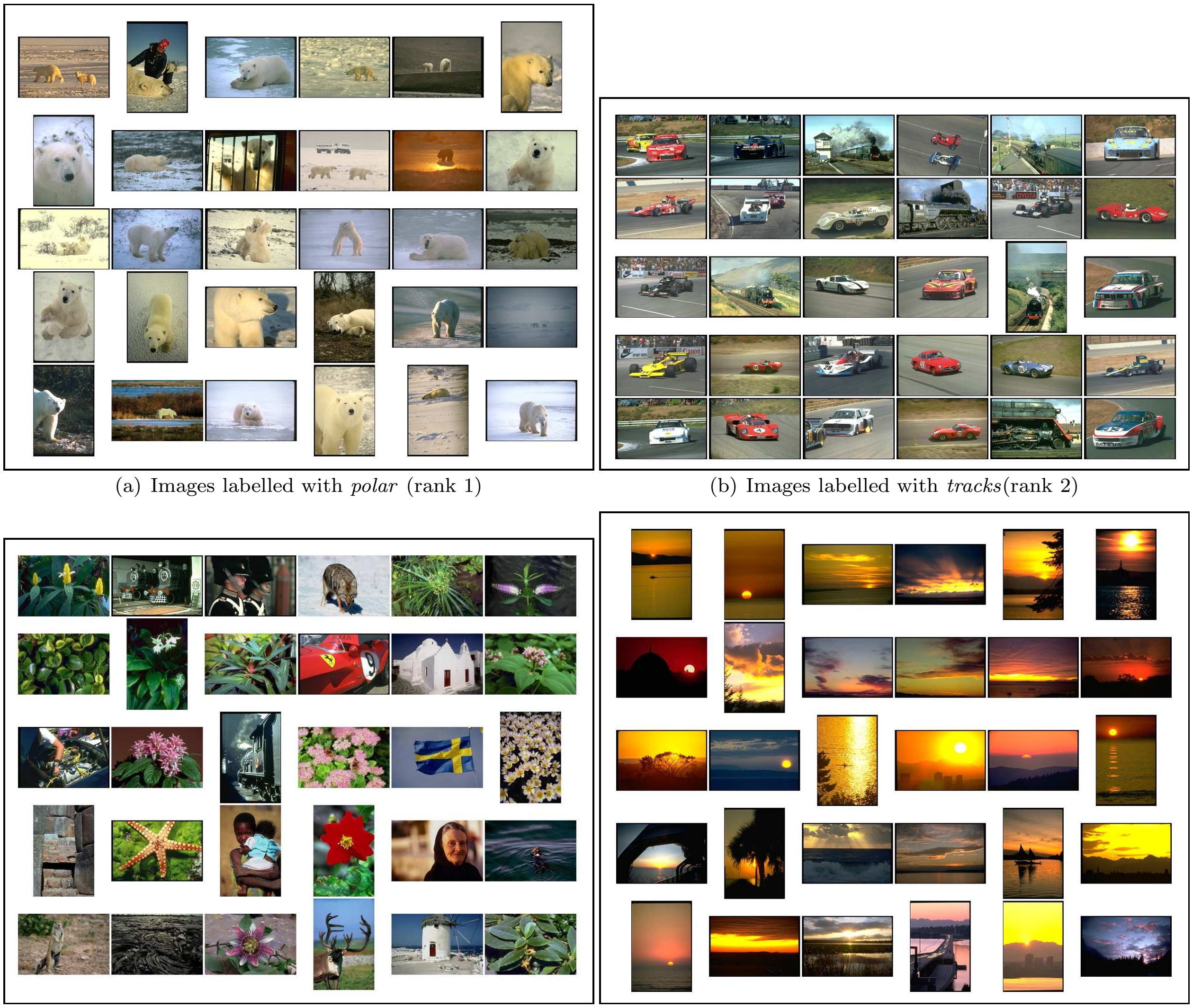
The benefit of scaling $\tilde{t}$ by $\tilde{u}$ is that the value c_w becomes comparable for different feature spaces that have the same dimensionality. Consider an extreme scenario when a particular feature always produces a random feature vector regardless of the input image. Scaling the median value of t in the above manner will result in every keyword sample receiving the same clustering value c_w of approximately 1, indicating that this feature representation is not useful for image annotation. In general this measure will tell us how much more densely T_w is clustered compared to a random sample of images. However, for the purposes of *ranking* the keywords, $\tilde{u}$ does not need to be computed and we can just use $\tilde{t}$. The median values are used for robustness to outliers in the samples. An important question is what number of nearest neighbours n should be used. One would expect that for larger samples the value $\tilde{t}$ will be smaller if n is kept constant, as the chances of sample points lying close to each other increase. A simple way to correct this bias is to set n to be a certain *percentage* of the sample size.

We would like to use this visualness measure to predict the accuracy of a probabilistic annotation model of a given keyword in a particular feature space. To this end we investigate the correlation of $\tilde{t}$ estimated on its training sample of a keyword with its average retrieval precision on the test set. We investigate how choosing the percentage value for n affects this correlation. Figure 4.5 illustrates the degree of this correlation on modified Corel, Corel, Getty and Corel-16k datasets, when the joint feature is used for annotation and n is set to be 1% of the keyword sample. The line in each plot shows a least-squares fit to the data. One can observe that in all four cases the correlation is negative, though this correlation is much weaker for Getty and Corel-16k datasets. To quantify these differences more precisely we calculated the correlation coefficients for different percentage values for n, shown in Table 4.2. The table shows that for all datasets there is negative correlation when using 1%, 2%, 5% and 10% of the sample size to define n. Smaller percentage values result in stronger correlation between visualness and average precision for all datasets. This indicates that a more localised measure of density is more effective for calculating $\tilde{t}$. The particularly low correlation of visualness and average precision on Getty and Corel-16k deserves special attention. By investigating Figures 4.5 (c,d) one can see that there are a large number of keywords which both have a high value of $\tilde{t}$ and high average precision. This suggests that although the feature vectors of these keywords are not tightly clustered, other keyword samples do not tend to occupy the same part of the feature space thus making accurate annotation possible in that particular evaluation setting. This scenario highlights the difficulty of designing an intrinsic measure of visualness of an image category that does not take other possible categories into account.

Tables 4.8 and 4.9 show modified Corel's keyword visualness rankings for under the joint feature and Log-Gabor alone, respectively. Some keywords appear near the top of the list for both features, such as *tracks*, *horses* and *plane*. Interestingly, other keywords differ in their relative visualness considerably, depending on the feature. For example, *sun* comes near the bottom for the joint feature but is near the top for Log-Gabor. To the contrary, *snow* ranks much lower for Log-Gabor than for the joint feature in terms of its visualness. Figure 4.1 shows example images from the two top ranking and the two bottom ranking modified Corel's keywords in terms of visualness under the joint feature. The images were randomly sampled from the collection for each keyword. The high visualness of *tracks* and *polar*, and the low visualness of *close-up* are intuitive. The bottom ranking of *sunset* is somewhat counter-intuitive, however, given the dominant colour cue in its images. One possible explanation is that these images have a lot of variability with respect to spatial colour layout which is what the MargCIE-T-3×3 part of the joint feature measures. Figure 4.2, likewise shows example images from the two top ranking and the two bottom ranking modified Corel's keywords in terms of visualness under Log-Gabor. One can see that the top ranking keywords are a lot less variable with respect to texture than the bottom two.

Table 4.10 shows top and bottom ranking Corel-16k's keywords in terms of visualness under the joint feature. Table 4.11 displays such information for the Getty collection and Table 4.12 for the Flickr dataset. In Corel-16k the most visual keywords under the joint feature tend refer to very specific sets of homogeneous images peculiar to the collection, e.g. *egg* (Figure 4.6 (a)) or to images of scenes that are visually coherent, e.g. *aerial*(Figure 4.6 (b)). This dichotomy is also true for the Getty collection: *Dessert* describes a homogeneous set of images while *Extreme Terrain* refers to a more general scene that appears visually coherent[3]. The Flickr collection is less likely to suffer from unnatural homogeneities, because it is a result of many different users uploading a few photos each. This is reflected in the visualness ranking of its tags: most tags ranked top relate to more visual scenes such as *fog, sand, rocks, mountains, beach, landscape, scenery, field* and *sea*. Corel-16k keywords are ranked bottom with respect to the joint feature tend to relate to objects and scenes with variable contexts e.g. *vegetable* (Figure 4.6 (c)) or to abstract words, e.g. *evening* (Figure 4.6 (d)). In Getty this pattern is also evident: keywords such as *Circle*, *Burning* and *Road Sign* amongst the least visual. The bottom ranking of the Flickr tags such as *sign, shadows, light, fire, lines, lamp* and *abstract*is consistent with this trend. Figure 4.7 shows example images of Flickr tags with high and low visualness under the joint feature.

[3] Unfortunately we are unable to supply example images for these keywords owing to copyright restrictions Getty Images have placed on the thumbnails we downloaded.



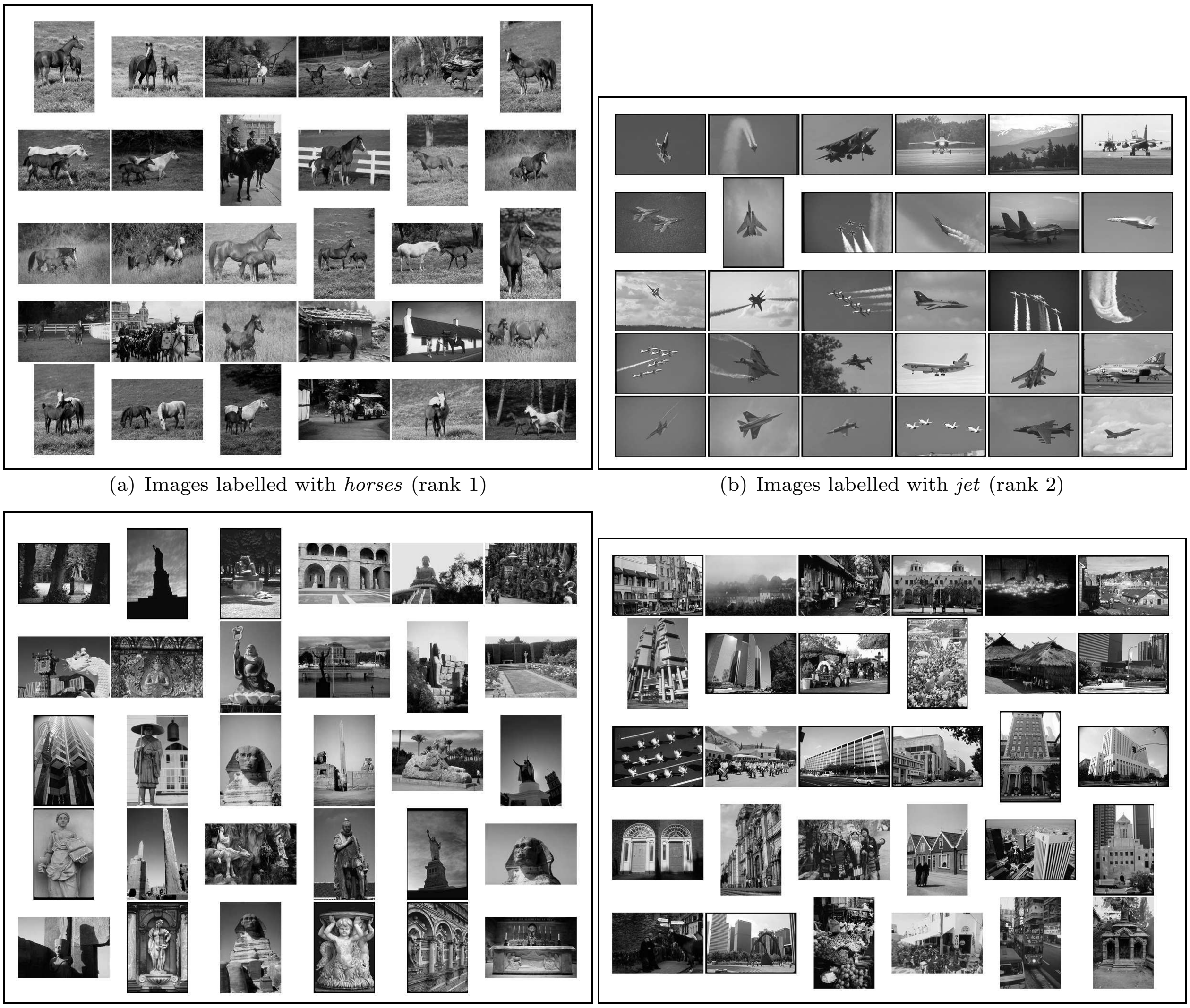
(a) Images labelled with *polar* (rank 1)

(b) Images labelled with *tracks*(rank 2)

(c) Images labelled with *close-up* (rank 35)

(d) Images labelled with *sun* (rank 36)

Figure 4.1: Keywords in modified Corel with respect to the joint feature.



(a) Images labelled with *horses* (rank 1)

(b) Images labelled with *jet* (rank 2)

(c) Images labelled with *statue* (rank 35)

(d) Images labelled with *street* (rank 36)

Figure 4.2: Keywords in modified Corel ranked with respect to Log-Gabor feature (no colour).

4.5 Keyword model generalisation performance

Finally we would like to see whether automatically annotating images from one dataset using keyword models estimated on another results in reasonable retrieval precision. In this section we use keyword models estimated on Corel-16k's training set to annotate Web images and modified Getty datasets. For every keyword w in the vocabularies of the latter two collections we identify the equivalent keyword w' in Corel-16k's vocabulary. The probability of keyword w for an image x is then defined as

$$p(w|x) = \frac{f(x|w')p(w)}{f(x)} \tag{4.1}$$

Dataset	Substituted density A.P.	Original density A.P.	Random
Web images	0.0534	0.3341	0.0074
Modified Getty	0.1200	0.2188	0.0397

Table 4.3: Effects of substituting Corel-16k keyword models for annotating Web images and modified Getty collections.

where $f(x) = \sum_w f(x|w')$. Note that w's prior probability, $p(w)$, remains unchanged; only the original density estimate $f(x|w)$ is substituted by Corel-16k's density function $f(x|w')$. Tables A.7 and A.8 in Appendix A show Corel-16k keyword associations for the modified Getty and Web image collections. When the identical keyword could not be found in Corel-16k to the keyword in question, a similar keyword was substituted instead (e.g. $w = Jug$, $w' = drink$ in Table A.8).

Table 4.3 summarises the effect of using substituted keyword models from Corel-16k on the two collections. In both cases the retrieval accuracy is significantly lower than when the collections' respective training sets were used for estimating keyword densities $f(x|w)$. This difference is particularly acute for the Web images dataset, where there is a six-fold drop in precision. On the other hand, in both cases the retrieval performance is an order of magnitude better than random retrieval. This result demonstrates that although there are profound differences between these datasets, the Corel-16k keyword models still contain relevant information for annotating the two collections – a result that we had hoped for. Tables A.9 and A.10 in Appendix A show how keyword model substitution affects retrieval precision of individual keywords in the Web image and modified Getty datasets, respectively.

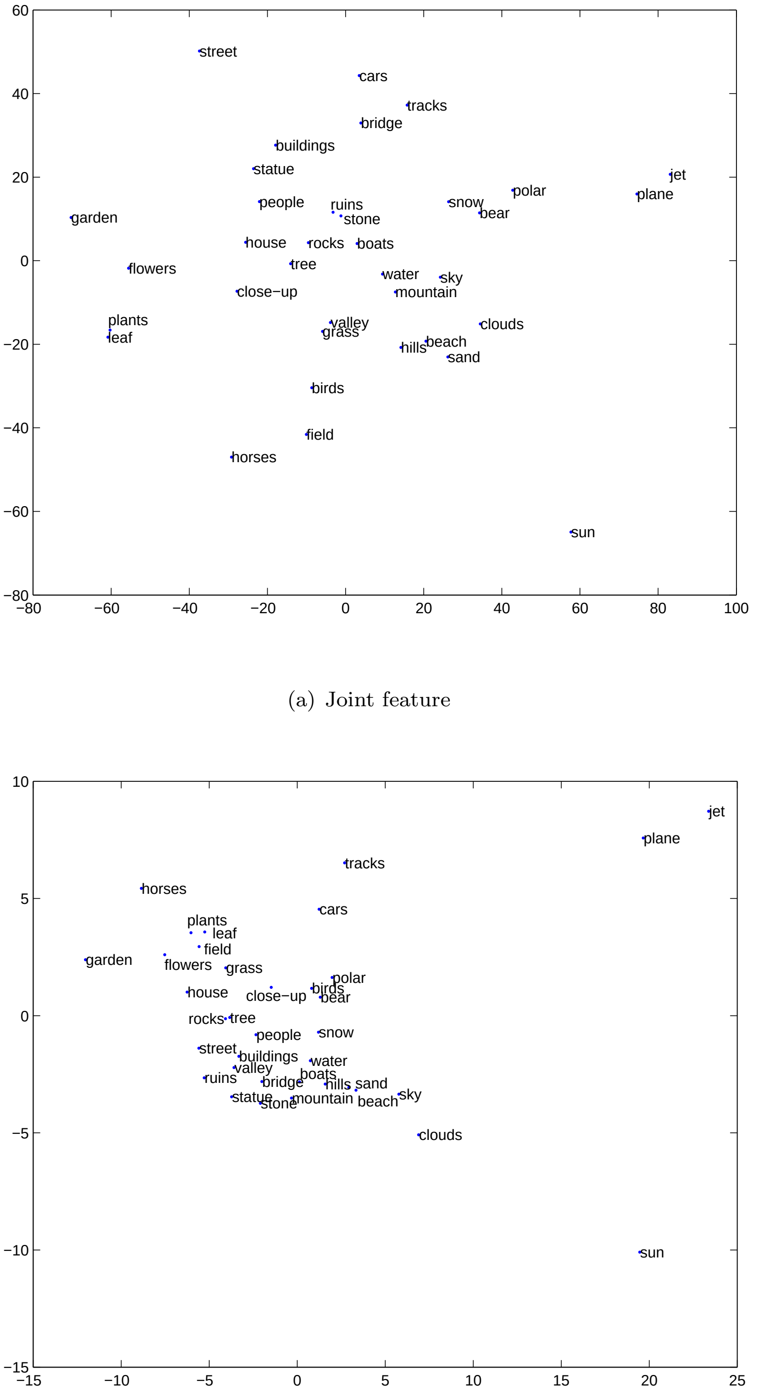
4.6 Conclusions

In this chapter we have investigated whether models of similar keywords have similar global feature distributions, even if models of these distributions were estimated on different datasets. The results are that

- in a significantly large number of cases, keywords relating to semantically similar concepts have similar low-level feature distributions within datasets
- in a significantly large number of cases, keywords relating to identical concepts in *different* datasets have similar feature distributions, and
- automatically annotating images from one dataset using keyword models estimated on another results in reasonable annotation performance, though significantly below that obtained by using the same dataset for both model estimation and evaluation.

These outcomes support our hypothesis that our global features are, in some cases, suitable for modelling real-world contextual dependencies in images that are relevant to the chosen concepts. Additionally, we have proposed a keyword 'visualness' measure which correlates with the keyword's model accuracy in our experiments. This measure could be used for removing keywords that would be unlikely to be modelled accurately with the chosen features.

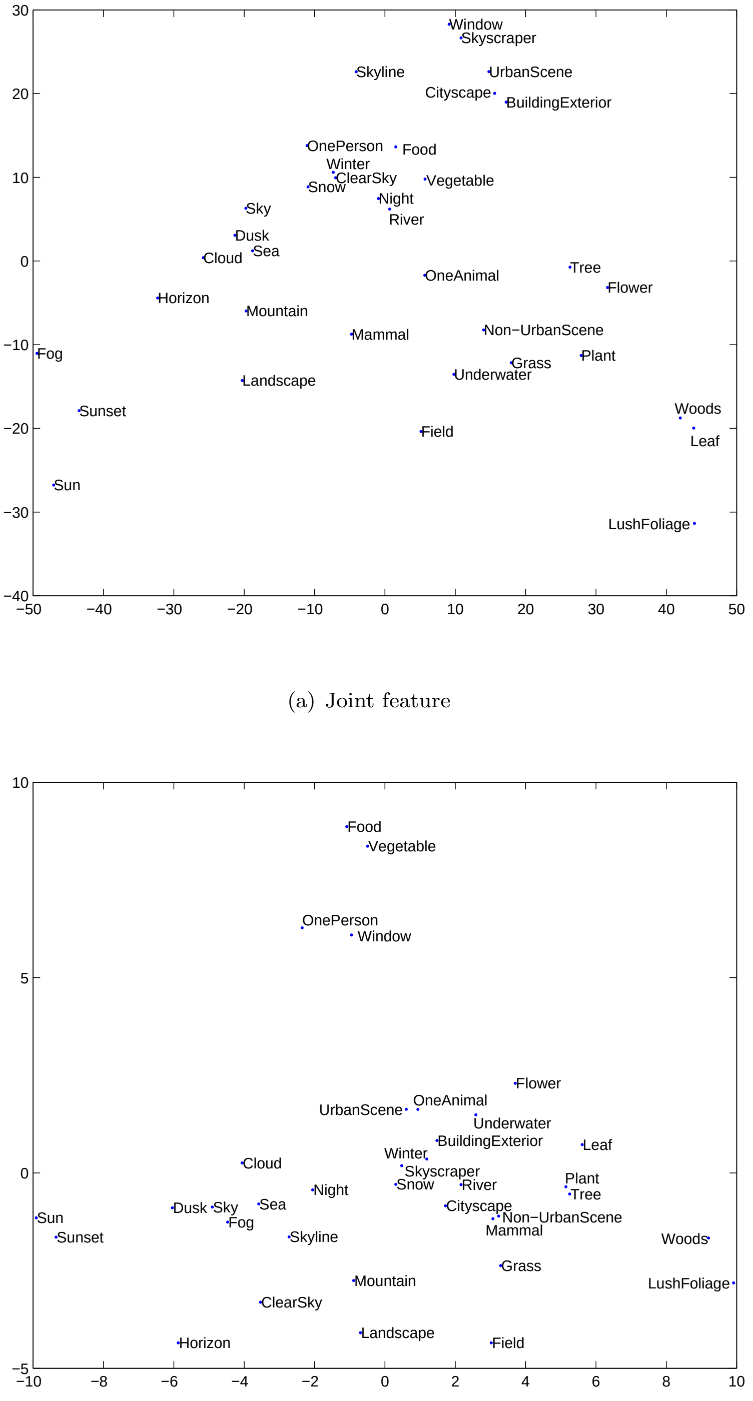
One interesting result is that using keyword samples from one dataset to annotate another can give significantly suboptimal results. In the next chapter we propose a technique for ameliorating this situation once such annotation has taken place.



(a) Joint feature

(b) Tamura-T-3×3

Figure 4.3: MDS of EMD distances between keyword feature distributions in the modified Corel dataset.



(a) Joint feature

(b) Tamura-T-3×3

Figure 4.4: MDS of EMD distances between keyword feature distributions in the modified Getty dataset.

Building Exterior
Urban Scene, *House*, Tree, *City*, *Structure*, *Cityscape*, Rock, Non-Urban Scene, *Town* (`Joint feature`)
Urban Scene, *House*, *Structure*, Tree, *Cityscape*, Non-Urban Scene, *City*, River, Rock (`Tamura-T-3×3`)
Flower
Plant, Tree, Summer, Non-Urban Scene, One Animal, Rural Scene, Animals In The Wild, Building Exterior, *Leaf*
Plant, Non-Urban Scene, Tree, Summer, Animals In The Wild, One Animal, Rural Scene, Autumn, *Leaf*
Landscape
Mountain, Cloud, Hill, *Horizon*, Extreme Terrain, Sky, Lake, Coastline, Sea
Hill, Rural Scene, Mountain, Field, Non-Urban Scene, Grass, Extreme Terrain, Sunlight, Cloud
Mammal
Animal, River, *One Animal* Rock, Snow, *Deer*, Mountain, Non-Urban Scene, Winter
Plant, Non-Urban Scene, Tree, Rural Scene, *Animal*, *Animals In The Wild*, *One Animal*, Grass, River
One Person
Head And Shoulders, Food, One Animal, *People*, *Men*, *Women*, Beach, Bed, Animal
Food, One Animal, *Men*, Water, Women, Urban Scene, Table, Animals In The Wild, *People*
Sea
Cloud, *Beach*, Sky, Mountain, *Coastline*, Horizon, *Water*, Snow, Lake
Cloud, *Beach*, Sky, *Coastline*, *Water*, Sand, Sunlight, Urban Scene, Mountain
Skyline
Harbor, Building Exterior, *Cityscape*, *Urban Scene*, *City*, Tower, Structure, *Sky*, *Clear Sky*
Harbor, *Cityscape*, Waterfront, Building Exterior, *Sky*, Urban Scene, Tower, Structure, *Clear Sky*
Sunset
Sun, *Dusk*, Silhouette, Cloud, Sunrise, Twilight, Dawn, Horizon, Mountain
Horizon, *Dusk*, Sea, Cloud, Sky, *Sun*, Silhouette, Beach, Sand
Underwater
One Animal, Animals In The Wild, Water, Animal, Non-Urban Scene, Plant, Sunlight, Rock, Tree
One Animal, Animals In The Wild, Animal, Plant, Mammal, Non-Urban Scene, Leaf, Water, Snow
Woods
Forest, *Tree*, *Lush Foliage*, *Rainforest*, *Plant*, Non-Urban Scene, Leaf, Grass, Autumn
Forest, *Lush Foliage*, *Rainforest*, Autumn, *Tree*, Plant, Crop, Non-Urban Scene, Leaf

Table 4.4: Within-dataset keyword distribution similarities for Getty Images. Words which the author judged as semantically similar to each keyword being considered are emphasised in italics.

building
architecture, *buildings*, *structure*, exterior, *castle*, ruins, stone, sky, *city* (`Joint feature`)
architecture, *buildings*, *structure*, exterior, water, ruins, sky, vegetation, *castle* (`Tamura-T-3×3`)
flower
plant, closeup, nature, *flora*, *leaves*, *vegetation*, fruit, *plants*, wildlife
flora, *plant*, closeup, *leaves*, nature, wildlife, animal, mammal, *vegetation*
landscape
water, scenic, sky, clouds, mountains, mountain, valley, scenery, grass
water, scenic, sky, vegetation, mountains, buildings, valley, clouds, mountain
mammal
animal, *wildlife*, rock, grass, rocks, *cat*, birds, vegetation, stone
animal, *wildlife*, nature, vegetation, leaves, grass, rock, tree, birds
person
people, animal, mammal, wildlife, water, rock, sky, closeup, stone
people, plant, flower, closeup, animal, nature, fruit, wildlife, mammal
sea
water, landscape, scenic, sky, clouds, island, scenery, mountains, rock
water, landscape, scenic, sky, sand, vegetation, buildings, rock, animal
skyline
city, *sky*, *buildings*, water, scenic, landscape, clouds, reflection, tower
city, *buildings*, *sky*, landscape, water, structure, scenery, scenic, *building*
sunset
dusk, *sun*, dawn, *twilight*, silhouette, *nightfall*, clouds, *evening*, landscape
dusk, *sun*, dawn, horizon, beach, sky, clouds, *twilight*, silhouette
underwater
fish, nature, closeup, vegetation, plant, leaves, wildlife, rock, animal
fish, leaves, nature, closeup, flora, detail, animal, rock, wildlife
woods
forest, *tree*, floor, *vegetation*, nature, park, autumn, stone, wall
forest, autumn, garden, floor, *plants*, moss, leaves, *vegetation*, wall

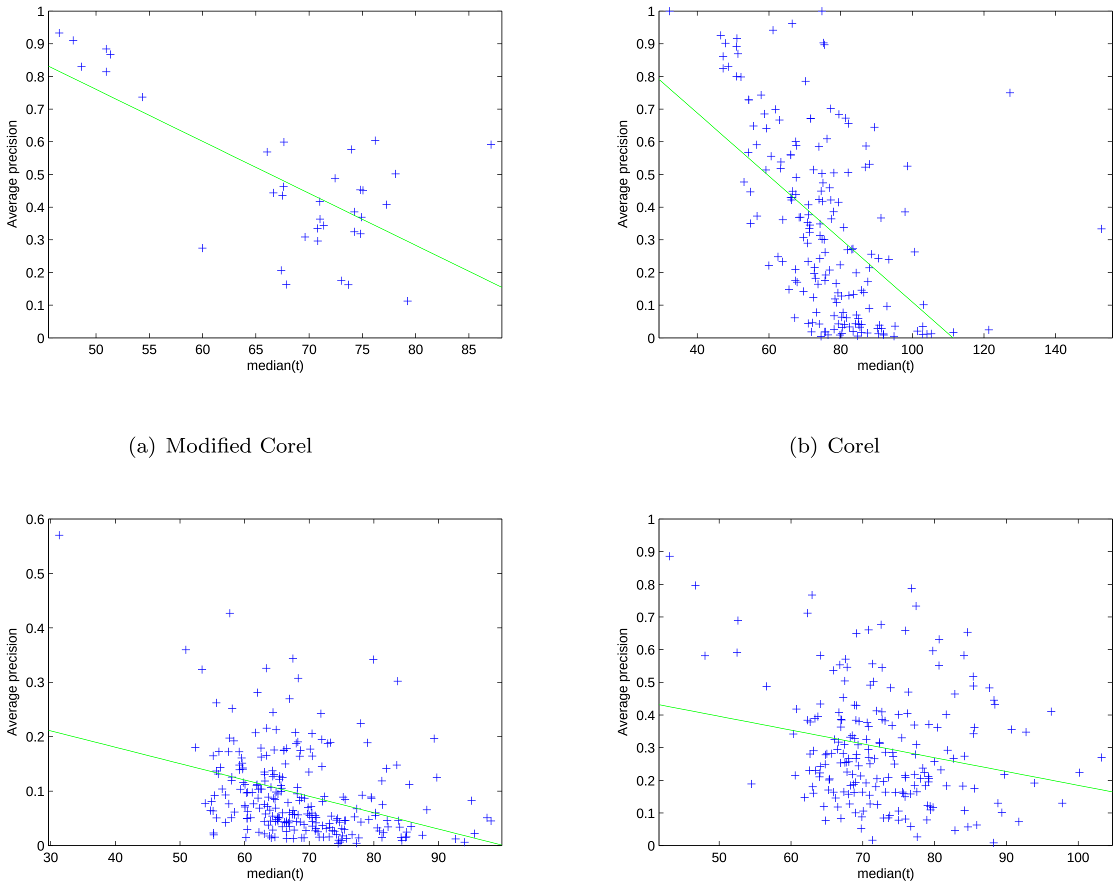
Table 4.5: Within-dataset keyword distribution similarities for Corel-16k. Words which the author judged as semantically similar to each keyword being considered are emphasised in italics.

Aerial view
water, grass, landscape, scenic, rocks, vegetation, river, park, rock (`Joint feature`)
landscape, clouds, water, scenic, valley, mountains, sand, sky, grass (`Tamura-T-3×3`)
Building exterior
architecture, **building**, buildings, ruins, structure, rocks, exterior, castle, stone
clouds, water, sky, landscape, **building**, architecture, mountains, scenic, sand
Flower
plant, **flower**, closeup, leaves, flora, nature, fruit, food, vegetation
clouds, sand, closeup, water, sky, office, room, reflection, plant
Jug
animal, landscape, mammal, water, sky, scenic, clouds, people, sand
clouds, beach, sand, water, sky, reflection, scenic, people, landscape
Mugshot
mammal, animal, people, clouds, landscape, water, scenic, sky, rock
clouds, beach, water, sand, scenic, sky, room, office, landscape
Clouds
cloudy, mist, **clouds**, sunset, landscape, horizon, wilderness, fog, beach
sunset, dusk, sun, cloudy, dawn, beach, **clouds**, horizon, nightfall
Crowd
tree, building, architecture, detail, rocks, buildings, nature, leaves, vegetation
clouds, architecture, water, building, sky, scenic, landscape, mountains, buildings
Grazing animal
grass, field, landscape, mammal, scenic, water, valley, vegetation, **animal**
landscape, clouds, field, mountains, grass, scenic, water, valley, sky
Mountain
landscape, clouds, **mountains**, scenic, valley, sky, hills, water, horizon
clouds, sky, horizon, landscape, scenic, island, beach, valley, hills
Sunset
sunset, dusk, sun, twilight, nightfall, dawn, clouds, cloudy, sunrise
sunset, nightfall, dusk, twilight, sun, dawn, horizon, cloudy, lighthouse
Underwater fish
water, landscape, scenic, sky, clouds, nature, animal, vegetation, wildlife
clouds, water, sky, landscape, scenic, sand, beach, mountains, reflection

Table 4.6: Web images/Corel-16k keyword distribution similarities. Matching keywords are highlighted in bold.

building
Building Exterior, Urban Scene, House, City, Structure, Cityscape, Tree, Statue, Street (`Joint feature`)
Food, Vegetable, Flower, Tomato, Bread, Dessert, Plate, Fruit, Women (`Tamura-T-3×3`)
flower
Flower, Plant, Animals In The Wild, Leaf, Animal Head, Autumn, Tree, One Animal, Branch
Food, **Flower**, Vegetable, People, Tomato, Fruit, School of Fish, Bread, Dessert
landscape
Rock, Building Exterior, River, Structure, Tree, Town, Urban Scene, Non-Urban Scene, City
Flower, Food, Vegetable, School of Fish, Animals In The Wild, Dessert, Plant, Tree, People
mammal
Building Exterior, One Animal, Urban Scene, Rock, Tree, House, Food, Stone, Structure
Food, Flower, Vegetable, Tomato, Dessert, School of Fish, Bread, Fruit, Hat
person
One Person, Women, Animal Head, Food, Men, Head And Shoulders, Table, One Animal, Window
Food, Tomato, Vegetable, Dessert, Bread, Head And Shoulders, Fruit, People, Women
sea
Rock, Urban Scene, River, Water, Building Exterior, Cityscape, City, Non-Urban Scene, One Animal
Flower, Food, Vegetable, Animals In The Wild, Dessert, Tomato, School of Fish, Bread, Fruit
skyline
Skyline, Cityscape, Urban Scene, City, Building Exterior, Tower, Harbor, Structure, Church
Lion, Animals In The Wild, Flower, One Animal, Animal, Fruit, Food, School of Fish, Dessert
sunset
Sunset, Dusk, Silhouette, Twilight, Cloud, Moody Sky, Sun, Dawn, Sunrise
Cloudscape, Food, Fruit, Cloud, Animals In The Wild, Vegetable, Storm Cloud, Dessert, Animal
underwater
Animals In The Wild, Plant, Flower, Tree, **Underwater**, Wildlife, Animal Head, One Animal, Urban Scene
Branch, Flower, Food, Tomato, Fruit, School of Fish, Animal Head, Christmas Tree, Vegetable
woods
Tree, **Woods**, House, Forest, Building Exterior, Plant, Urban Scene, Stone, Branch
Woods, Waterfall, Autumn, Rainforest, Forest, Flower, Tree, Lush Foliage, Plant

Table 4.7: Corel-16k/Getty Images keyword distribution similarities. Matching keywords are highlighted in bold.



(a) Modified Corel

(b) Corel

(c) Getty

(d) Corel-16k

Figure 4.5: Keyword average precision on the test set versus keyword visualness $\tilde{t}$ on the training set; $n = 1\%$.

Keyword	$\tilde{t}$
polar	46.54
tracks	47.85
horses	48.64
plane	50.96
bear	50.96
jet	51.35
cars	54.35
ruins	59.97
snow	66.06
field	66.64
garden	67.39
birds	67.49
grass	67.61
stone	67.64
rocks	67.86
sand	69.62
mountain	70.80
bridge	70.81
beach	71.00
house	71.02
clouds	71.38
valley	72.44
hills	73.02
boats	73.69
water	73.96
leaf	74.24
plants	74.24
flowers	74.80
statue	74.82
tree	74.92
buildings	75.06
other	75.72
sky	76.21
street	77.27
people	78.12
close-up	79.25
sun	87.07

Table 4.8: Modified Corel's keyword visualness for the joint feature

Keyword	$\tilde{t}$
horses	10.66
jet	12.54
sun	12.81
plane	12.86
tracks	13.13
polar	13.96
field	14.14
cars	14.31
bear	14.50
sand	14.53
beach	14.75
clouds	15.67
valley	15.84
ruins	16.08
mountain	16.10
leaf	16.28
grass	16.31
hills	16.35
stone	16.51
garden	16.57
house	16.67
boats	16.91
plants	16.92
birds	17.00
water	17.30
tree	17.32
rocks	17.33
sky	17.39
flowers	17.50
snow	17.76
other	18.52
people	18.92
buildings	19.04
close-up	19.42
bridge	19.97
statue	20.04
street	20.43

Table 4.9: Modified Corel's keyword visualness for Log-Gabor

Rank	Keyword	$\bar{t}$
1	egg	43.12
2	duck	46.70
3	decoration	48.03
4	pebbles	52.48
5	museum	52.61
6	antelope	54.49
7	frost	56.61
8	castle	60.32
9	lion	60.59
10	waterfall	60.76
11	hills	61.91
12	train	62.30
13	scenery	62.31
14	exterior	62.56
15	aerial	62.66

Rank	Keyword	$\bar{t}$
189	twilight	87.62
190	factory	88.22
191	textile	88.28
192	sun	88.40
193	leaf	88.81
194	spring	89.37
195	sign	89.70
196	scene	90.72
197	meadow	91.73
198	glass	92.74
199	sunrise	93.89
200	evening	96.24
201	vegetable	97.78
202	metal	100.18
203	silhouette	103.25

Table 4.10: Top and bottom ranking Corel-16k keywords under the joint feature

Rank	Keyword	$\bar{t}$
1	Wolf	31.32
2	Deer	50.91
3	Lion	52.38
4	Mammal	53.41
5	Town	53.91
6	Dessert	54.74
7	Village	54.90
8	Surf	55.13
9	Woods	55.14
10	Carving	55.19
11	Canyon	55.22
12	Extreme Terrain	55.36
13	Rainforest	55.63
14	Hill	55.64
15	Bathroom	55.66

Rank	Keyword	$\bar{t}$
233	Multiple Lane Highway	84.89
234	Circle	85.02
235	Water Surface	85.08
236	Computer Monitor	85.47
237	Sign	85.75
238	Burning	87.50
239	Suspension Bridge	88.18
240	Petal	89.33
241	Single Flower	89.73
242	Railroad Track	92.65
243	Road Sign	94.02
244	American Flag	95.10
245	Arrow Sign	95.58
246	Flag	97.49
247	Cable	98.11

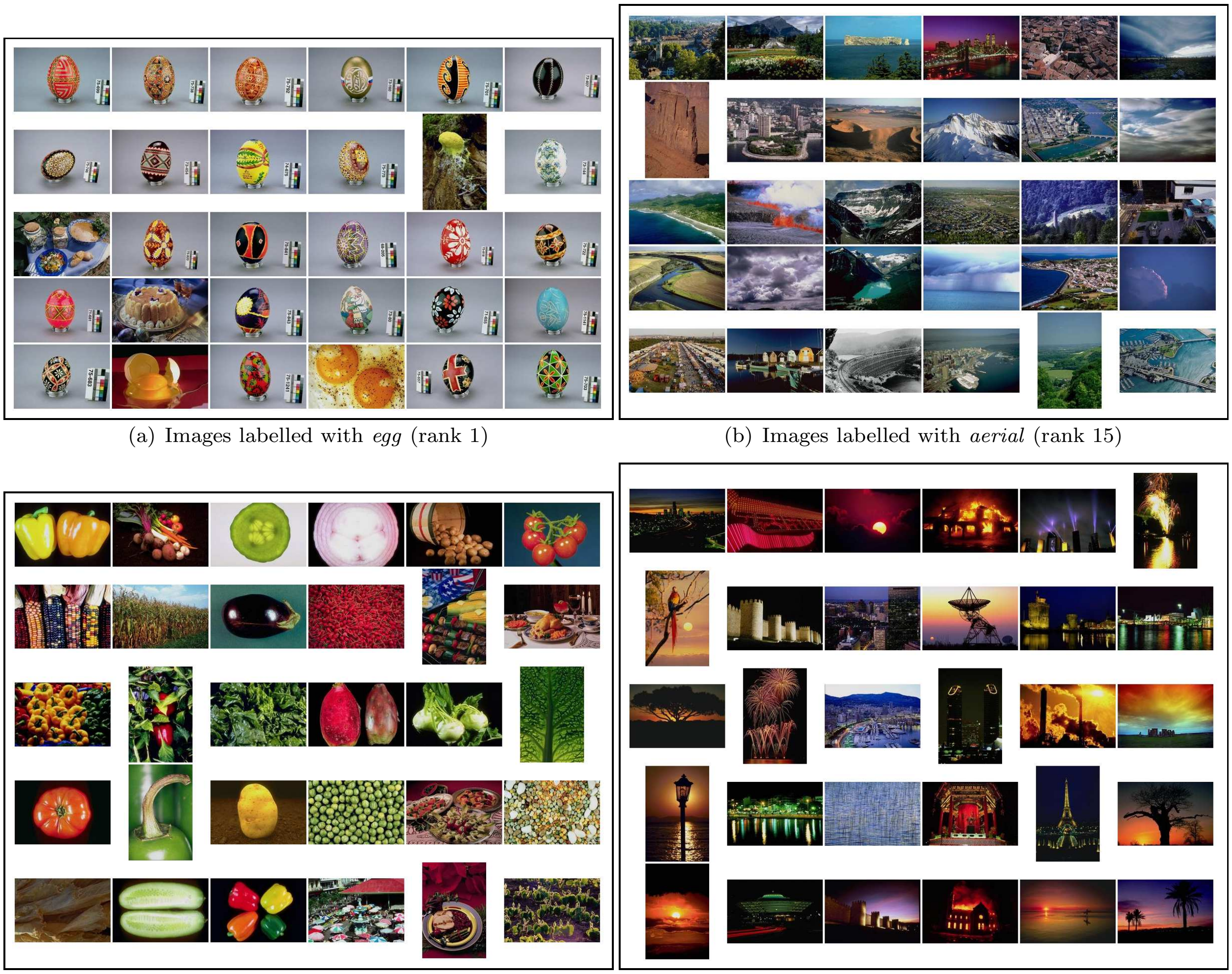
Table 4.11: Top and bottom ranking Getty keywords under the joint feature

Rank	Keyword	$\bar{t}$
1	fog	53.49
2	sand	54.40
3	rocks	55.35
4	mountains	55.67
5	mountain	57.52
6	baby	57.92
7	snow	58.08
8	beach	58.08
9	horse	58.57
10	landscape	59.03
11	scenery	59.17
12	island	59.32
13	stone	59.60
14	field	59.75
15	sea	59.83

Flickr keywords

Rank	Keyword	$\bar{t}$
159	sign	77.75
160	shadows	77.90
161	toy	78.07
162	bar	78.61
163	light	78.86
164	flag	79.45
165	fire	80.43
166	yellow	80.96
167	lines	81.54
168	orange	82.01
169	bright	82.58
170	red	83.17
171	glow	83.88
172	lamp	85.36
173	abstract	86.83

Table 4.12: Top and bottom ranking Flickr keywords under the joint feature



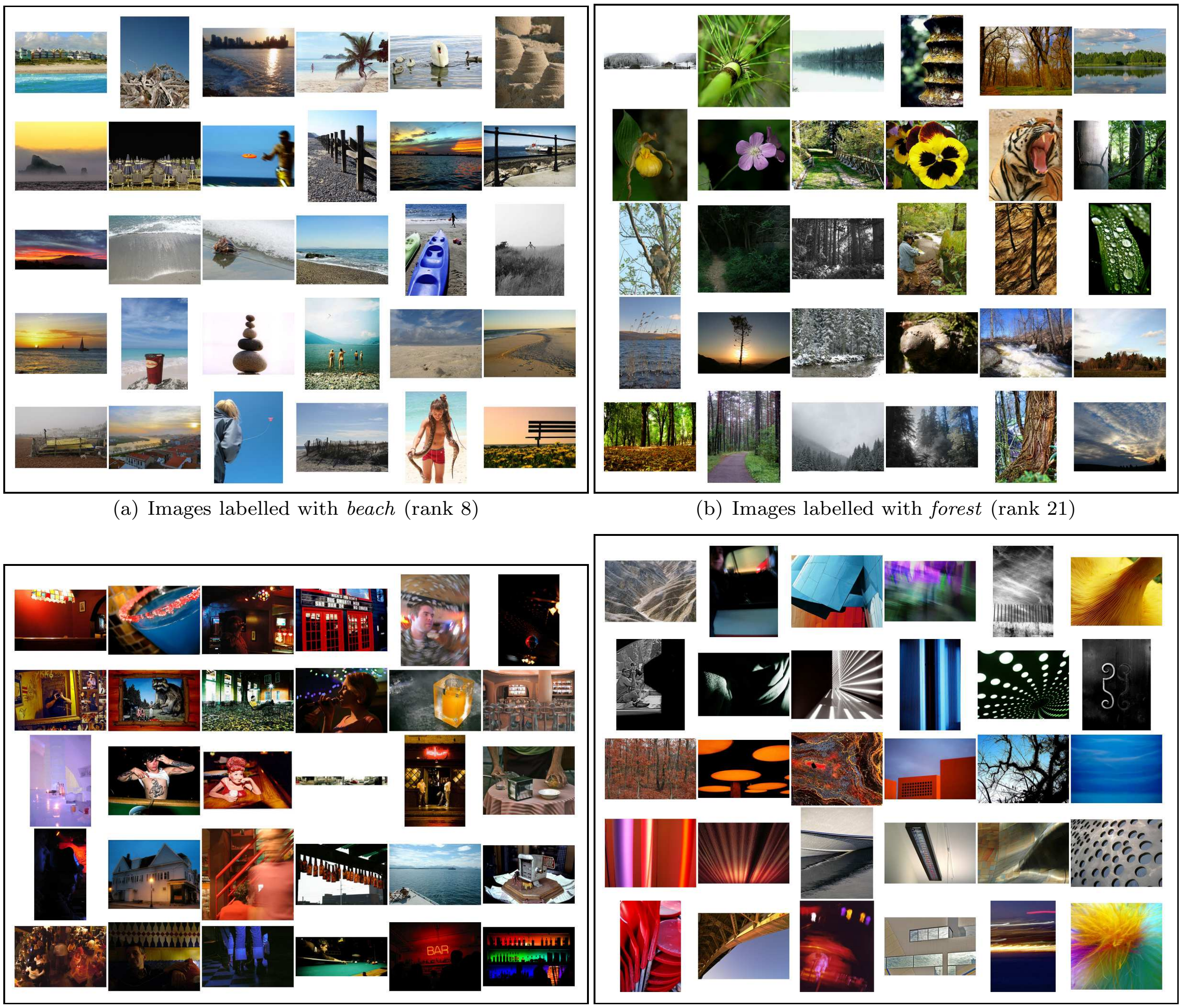
(a) Images labelled with *egg* (rank 1)

(b) Images labelled with *aerial* (rank 15)

(c) Images labelled with *vegetable* (rank 201)

(d) Images labelled with *evening* (rank 200)

Figure 4.6: Keywords in Corel with visualness rankings with respect to the joint feature.



(a) Images labelled with *beach* (rank 8)

(b) Images labelled with *forest* (rank 21)

(c) Images labelled with *bar* (rank 162)

(d) Images labelled with *abstract* (rank 173)

Figure 4.7: Tags in Flickr with visualness rankings with respect the joint feature.

Chapter 5

Efficient Re-indexing of Automatically Annotated Images

In the previous chapter we investigated the generality of the proposed method for estimating image keyword probabilities. We have learned that annotating images from one datased using keyword models estimated on another can often result in inferior performance compared to using the same dataset for both training and evaluation. In practice this condition is likely to arise quite frequently. It is easy to imagine a situation where it is impossible to obtain sufficient training samples from the image collection that is about to be automatically annotated. A different image dataset would then have to be used for estimating keyword models. On the other hand, it is also conceivable that manually labelled images gradually become available in the former collection after the annotation process has been performed (e.g. through user feedback). The new labels may correspond to keywords in the original annotation vocabulary or may relate to entirely new concepts. In both cases it would be appealing to express this newly acquired knowledge using the pre-computed automatic annotations.

As we have seen in the previous chapter, images of similar concepts are often distributed similarly in low-level feature space, even when considering different datasets. This result suggests that when the accuracy of individual keyword probabilities is poor there may still be a significant amount of useful information present in the automatic annotations. Based on this hypothesis we propose a *re-indexing* framework that aims to support refinement and augmentation of the annotation vocabulary at low computational cost. Within this framework, pre-computed keyword probabilities are re-used for improving retrieval of concepts that are not modelled accurately by corresponding individual keywords. They are also used for labelling images with new concepts which are not present in the annotation vocabulary. This is achieved by automatically identifying a small set of keywords which discriminate best between images that contain the given concept and those which do not. The basic idea behind this approach is to find a set of keywords, the combined probabilities of which describe the given concept better than the probability of any individual keyword in the annotation vocabulary. We would hope that only a few keywords would be required if models of their distributions in low-level feature space carry enough useful information for the given task.

A somewhat light parallel can be drawn between our approach and *automatic query expansion* in text information retrieval. Automatic query expansion is a relevance feedback method which improves recall by inserting related query terms into the user's query (Rocchio, 1971; Robertson and Sparck-Jones, 1976). Additional terms are selected from documents matching the user's information need; these documents are either selected manually or are defined to be the top n documents returned in response to the original query (the latter approach is usually referred to as blind relevance feedback (Buckley et al., 1994; Mitra et al., 1998). This automatic query reformulation process happens at search time and capitalises individual user feedback, or – in the case of blind relevance feedback – uses no explicit feedback at all. In contrast, our re-indexing technique is intended to be performed offline which would allow one to utilise long-term relevance feedback from multiple users.

Our approach is more conceptually similar to another text retrieval technique – *automated text categorisation* – concerned with constructing a classifier that correctly groups documents into two or more predefined categories. Such a classifier is trained on a manually categorised document collection and its output is typically a weighted combination of different term frequencies (for a good overview of text categorisation the reader is referred to Sebastiani (2002)). For example, Joachims (1998) classifies text documents into semantic categories by using documents' term-frequency vectors in conjunction with the Support Vector Machine classifier. Our goal is more specific: we would like the re-indexing procedure to be computationally cheap compared to performing image annotation from scratch using the newly labelled images. Of course, we would also like our procedure to maitain similar levels of accuracy to the latter alternative, although a reasonable trade-off would be acceptable. Perhaps the closest text categorisation technique is that developed by Baker and McCallum (1998). The authors cluster words into groups based on the distribution of class labels associated with each word. This aggressively compresses the original feature space and results in very fast document classification, while maintaining high classification accuracy.

In this chapter we present a simple, efficient, and robust algorithm for keyword selection and contrast it to the Support Vector Machine classifier adapted for this task. Both keyword combination approaches are then compared to modelling each concept using a dedicated annotation model and differences in their respective accuracies are reported. To illustrate the generality of this approach, we state our retrieval results on Getty and Web image collections, both of which are significantly different to the Corel dataset that was used for training keyword models.

5.1 Related work

The idea of combining concept detector outputs in image retrieval has been studied before, particularly within the TREC Video Retrieval Evaluation (TRECVID) community (Naphade et al., 1998; Naphade and Huang, 2000; Smith et al., 2003; Natsev et al., 2003; Amir et al., 2003; Hauptmann et al., 2004; Wu et al., 2004; Yan et al., 2006; Rasiwasia et al., 2006). Before discussing related work, let us briefly outline the idea behind TRECVID. The aim of the yearly TRECVID conference is to promote progress in content-based retrieval from digital video via a metrics-based evaluation[1]. In the past, the central part of this exercise has been a laboratory-style evaluation of automated tasks on news video footage, such as shot boundary determination and detection of concept presence in videos, e.g. anchor-person detection. The latter task is similar to that of automated image annotation. However, we have not used TRECVID datasets for evaluating the techniques we propose in this thesis. We believe that modelling visual content of news video footage lies within a different problem domain to that of automated photograph annotation. Nonetheless, participants of the TRECVID effort have employed conceptually similar techniques to the one we propose in this chapter. We shall now outline these techniques and contrast their goals with those of our re-indexing framework.

Naphade et al. (1998); Naphade and Huang (2000) express relationships between concept classifiers in video keyframes using a Bayesian network, with the aim of modelling higher-level semantic classes of such keyframes. Smith et al. (2003) perform query-by-example retrieval with images represented by vectors of different concept classifier outputs and this method proves effective for a range of image queries on the TRECVID dataset. Natsev et al. (2003) exploit concept classifier dependence in video keyframes to construct new classifiers for concepts with insufficient numbers of training examples. Hauptmann et al. (2004) combine concept predictions using a logistic regression classifier, whilst Amir et al. (2003) do so with a Support Vector Machine. Wu et al. (2004) attempt to model relationships between concepts based on a predefined ontological hierarchy. Finally, Yan et al. (2006) use a range of graphical models for representing concept relationships to enhance concept detection. The goal of our approach radically differs to those in this body of work in the following ways. Instead of seeking to improve query-by-example or concept detection performance, our aim is to provide a computationally cheap alternative for refining existing concepts and modelling ones that do not exist in the training vocabulary. We do so by selecting a small number of keywords that can represent the concept of interest with a reasonable tradeoff in accuracy compared to a dedicated, annotation model. Importantly, we explicitly investigate situations where the training collection from which annotation models are estimated is *substantially different* to the collection that is being indexed. We believe that in such cases it is most likely that the quality of initial annotations will need to be improved over time.

[1] http://www-nlpir.nist.gov/projects/t01v/

5.2 Re-indexing framework

Initially the test collection is indexed using nonparametric keyword models described in Chapter 3. These models are trained on images from an unrelated dataset. Subsequently, given a set of images from the test collection which turn out to be relevant to a particular concept, we automatically choose a small set of keywords which, when combined, discriminate best between the above images and images which are irrelevant to that concept. There are two possible situations when this technique can be used:

The target concept has a corresponding keyword in the vocabulary. In this case we are using keyword combination in an attempt to improve retrieval performance of this concept as compared to using a single keyword.

The target concept has no corresponding keywords in the vocabulary. In this scenario we are using keyword combination as a substitute to training a dedicated model for the new concept and using it to annotate the rest of the test collection.

We next give details of our keyword combination methods and their computational cost. We end this section with a formal description of our experimental procedure used for evaluating our approach.

5.2.1 Models for keyword combination

Greedy keyword multiplication. Suppose that we ask an annotator to repeatedly assign a new keyword n times for an image x. The probability of the event that keywords $w_1, \dots, w_n$ are selected can be modelled as

$$p(w_1, \dots, w_n|x) = \prod_{i=1}^{n} p(w_i|x). \tag{5.1}$$

The above equation exploits the assumption that all keywords are assigned independently. The goal of the model is to find a set of keywords for which the average precision of a given concept is maximised when images are ranked according to Equation (5.1). We solve this problem by fixing the number of keywords, n, and using a greedy search algorithm that, for a given concept, repeatedly adds the keyword which produces the largest increase in average precision on the training set, until the desired number of keywords are inserted into the product.

Greedy keyword addition. It is also possible to model the event that given an image x the annotator will pick a keyword from a set W of n different keywords. The probability of this event is

$$p(W|x) = \sum_{w \in W} p(w|x). \tag{5.2}$$

For this model we likewise use greedy search to select n keywords which maximise the concept's average precision once images are ranked according to Equation (5.2).

The combinatorial interpretation of the above two approaches is that we are choosing n keywords from a vocabulary of size m *without* replacement. This results in $\binom{m}{n}$ potential solutions – typically a very large number to be evaluated on the training set. Greedy search is therefore used as a primitive heuristic.

Linear combination model. Here we take a different view to the above two models and use the automatically generated keywords within the standard classification framework. We aggreate the keyword probabilities of each image into a vector v and use a Support Vector Machine (SVM) to find a linear hyperplane that discriminates well between keyword vectors of images which contain a particular concept and of those which do not. This is similar to the work of Joachims (1998) in which term vectors of text documents are classified with an SVM, and to that of Amir et al. (2003) and Hauptmann et al. (2004), as described in Section 5.1.

SVMs, introduced by Vapnik (1995), are learning machines that are capable of performing binary classification. Given a set of l training points belonging to two separate classes the objective of the SVM is to separate them with a hyperplane function $\langle w, v\rangle + b = 0$ such that

$$\min |\langle w, v_i\rangle + b| = 1,$$

subject to the constraint

$$y_i[\langle w, v_i\rangle + b] \geq 1.$$

This specifies that the hyperplane must separate the two classes correctly with the maximum margin possible. The solution to this problem is found by the minimisation of the function $\frac{1}{2}\|w\|^2$ subject to the above constraints, which can be solved using quadratic programming.

The SVM is trained on keyword vectors to derive the hyperplane that separates the positive and the negative examples with least error. Once trained, the relevance score of an unseen image is defined as the distance of its keyword vector v to the hyperplane, which is just a linear weighted sum of the keyword vector's components offset by the constant factor b. In this context, the hyperplane represents the set of weights for the keywords that minimise the error on the training sample.

SVMs are known to have favourable generalisation properties compared with many other binary classifiers and fast implementations are widely available. One such implementation - Joachims (2001) - is used for our experiments.

Treating keyword probability vectors as visual features. It is also possible to consider the keyword vector v, introduced above, as a visual feature and treat it like simple global features in Chapter 3. In this context the above three methods combine feature selection and classification in this 'semantic' feature space. We can establish the relative effectiveness of such feature selction by applying the same nonparametric annotation models over the entire vectors and evaluating their accuracies. This performs re-indexing of images by using all keyword probabilities, and – in this context – does not throw away any information.

5.2.2 Computational complexity

The computational cost associated with labelling images using the keyword combination methods is slight compared to doing so with the concept-specific dedicated nonparametric annotation model described in Chapter 3. Greedy keyword combination requires only n multiplication or summation operations per image, respectively, where n is the number of keywords we wish to choose. The linear combination model requires m summation operations, where m is the number of nonzero elements in the hyperplane vector w. By contrast, the cost of the nonparametric model grows linearly in the number of training examples and in the dimensionality of the low-level feature vectors, as noted earlier. Both greedy and linear combination approaches have inexpensive parameter estimation procedures.

5.2.3 Performance evaluation

We use the following formal experimental procedure to evaluate our framework. We are asked to index a collection of images, A, in which images are manually labelled with concepts from vocabulary W_A (but which are assumed to be unobservable at the time of indexing). We annotate the entire collection using the nonparametric annotation model, described in Chapter 3, trained on a *reference collection* B, with its respective vocabulary W_B. A is then split into training and test sets, A_{train} and A_{test}, respectively. For each concept in W_A, keyword combination models outlined in Section 5.2.1 are trained on A_{train} and the resulting combinations of keywords picked from W_B are evaluated on A_{test}. Accuracies obtained on A_{test} (defined as average precision values) are averaged across all concepts in W_A. This simulates the process of re-indexing the collection using keyword combination for every concept in W_A. We also train a dedicated, nonparametric annotation model on *low-level features* of A_{train} and use it to annotate A_{test} directly with concepts from W_A. This approach serves as the *quasi-upper bound*. It shows how accurately it is possible to re-index A_{test} by starting from scratch instead of re-using existing keyword probabilities. By comparing our model accuracies to the upper bound we will be able to establish the relative effectiveness of our keyword combination strategies. An important condition for assessing the generality of our approach is that the images in the reference collection come from a different source to those which are in the collection we are trying to index. This is meant to prevent any unanticipated overfitting of keywords from W_B to concepts in W_A. We observe this requirement in our experiments.

Why is the nonparametric density estimate a quasi-upper bound? This is a subtle but an important point in our work. In principle, there is nothing fundamentally limiting the precision of keyword combination to be lower than that of a nonparametric density estimate of the target concept in low-level feature space. However, the keyword probabilities are estimated in the same feature space but on a different dataset to the one being annotated. It is therefore unlikely that combining a small number of them will model the distribution of the target concept in the low-level feature space as accurately as the nonparametric estimate. This is acceptable as our aim is to achieve a significant saving in computational effort at a reasonable performance tradeoff.

5.3 Experimental results

5.3.1 Image data and low-level features

We use the notation provided in Section 5.2.3 to describe our experimental setup. We use two datasets as collections A to be indexed: modified Getty and Web images, described in Chapter 3. In each case partitions A_{train} and A_{test} are simply the training and test partitions, previously used for our experiments, and W_A is the respective vocabulary. The reference collection B is the training partition of the large Corel dataset compiled in the previous chapter, with its corresponding vocabulary W_B. In words this means that we are using Corel to annotate Web images and the modified Getty collection. All models are estimated and evaluated in the combined feature space **MargCIE-3x3+Tamura-3x3+Log-Gabor**.

5.3.2 Annotation performance

We report the mean average precision of our keyword combination methods in Table 5.1. By default, 10 keywords are used for each concept by the greedy addition and multiplication methods. Greedy multiplication consistently outperforms the two other combination methods. Combining 10 keywords this way recovers 65% of the upper bound accuracy on Web images and 78% on the modified Getty dataset. The accuracy of the linear combination model is similar to greedy multiplication on Web images, but surprisingly is worse than both multiplication and addition on the modified Getty dataset. The accuracy of greedy addition is significantly lower than that of the multiplication method on Web images, and somewhat lower on the modified Getty collection. All reported figures are significantly above random chance.

It is interesting to look at the accuracy of the nonparametric annotation model that uses keyword probability vectors as visual features, also shown in Table 5.1. The Laplace kernel was used as before, and bandwidth was likewise optimised with respect to accuracy on the withheld evaluation set. It appears that greedy multiplication of just 10 keywords results in virtually the same performance as using entire probability vectors. This highlights the effectiveness of greedy multiplication from a feature-selection point of view.

	Dedicated nonparametric models	10 keyword product	10 keyword sum	SVM linear combination	Annotation using keyword vectors as visual features	Random retrieval
Web images	0.3341	0.2127	0.1342	0.1783	0.2344	0.0073
Getty	0.2188	0.1718	0.1561	0.0997	0.1644	0.0403

Table 5.1: Keyword combination mean average precision.

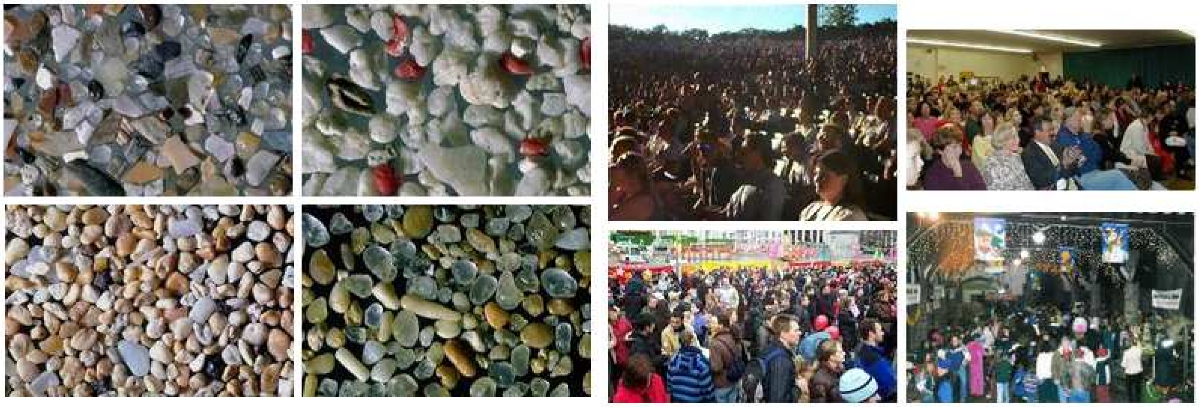
	Product	NPDE
Training	$51.2s$	$0s$
Annotation	$0.2s$	$8613.2s$
Total	$51.4s$	$8613.2s$

Table 5.2: CPU time taken to annotate the Web collection with one concept (seconds): greedy multiplication re-indexing vs. nonparametric density estimate. Measured on an Intel Pentium 4 3.00 GHz. System implemented on Sun's Java 1.5 platform

Overall, the results are encouraging – multiplying only 10 keywords is sufficient for obtaining between 60% and 80% of the quasi-upper bound performance on average. Table 5.2 compares the computational cost of re-indexing the Web collection with one concept using greedy multiplication, to that of the nonparametric density estimate. 10,000 images are used for training and 59,814 are annotated automatically, as before. The table shows that having this performance trade-off allows us to re-index the above collection over 100 times faster. Selecting keywords for multiplication in a greedy fashion takes most of the time – annotation itself requires only 9 multiplication operations per image. In constrast, the annotation cost of the nonparametric model grows linearly in the number of training examples and in the dimensionality of the low-level feature vectors.

Increasing the number of combined keywords improves the accuracy of both greedy search methods on both datasets, as Figure 5.2 shows, though the improvement for the multiplication method appears to be greater. For example, we note a two-fold accuracy improvement when 10 keywords are chosen to be multiplied instead of just one on the Web images. One can see that on average there is less of an increase in precision for the modified Getty dataset as more keywords are multiplied. The reason for this could be that the feature distributions of Corel keywords model the Getty concepts more closely, and thus there is relatively less to gain by combining keyword probabilities in this manner. Tables 5.3 and 5.4 show keyword combination precision details for each concept in the modified Getty and Web image datasets, respectively. In Table 5.3, concepts for which keyword multiplication results in significant precicion increase are marked with an asterisk ($*$).

It is also interesting to observe which keywords are selected by the greedy methods. Tables 5.6 and 5.5 show the keywords picked by the multiplication method for each of the concepts in both datasets. Note that for some concepts the corresponing keyword is not selected as the first one by the greedy search; this shows that it may be suboptimal to use keywords that match target concepts because our keyword models were estimated on a different dataset. It appears that the selected keywords are sometimes irrelevant to the target concept. However, since the probabilities of these keywords are estimated using low-level image features, an ‘irrelevant’ keyword may well be helpful in characterising a particular visual aspect of that concept. An extreme example is the use of the keyword *pebbles* from Corel to describe the *Crowd* concept in Web images; Figure 5.1 illustrates why it might have been picked by the greedy search algorithm.



(a) Images labelled with *pebbles* in Corel

(b) Images labelled with *Crowd* in Web image dataset

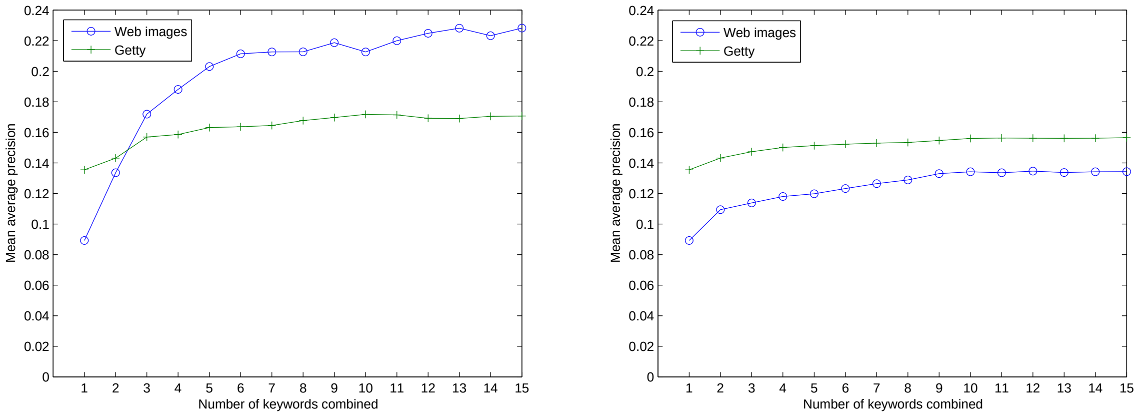
Figure 5.1: Note the texture similarity between *pebbles* and *Crowd*

.

Figures 5.5 (a), 5.6 (a), 5.7 (a) and 5.8 (a) show how multiplying more keywords improves test average precision for Getty’s *Clear Sky* and *River* and Web image dataset’s *Grazing animal* and *Aerial view* concepts, respectively. Figures 5.5 (b), 5.6 (b), 5.7 (b) and 5.8 (b) show the same relationships for the training average precision of respective concepts. Each labelled point in a graph corresponds to another keyword picked by the greedy algorithm. One can observe that the training average precision curves are generally much smoother than the test ones – an expected finding.

Our approach has the capability of retrieving concepts which are not present in the vocabulary of the reference collection. The Web dataset has three such concepts: *crowd*, *jug* and *mugshot*. Table 5.4 shows that for all three concepts the accuracy of the greedy multiplication approach is substantially lower than the nonparametric quasi-upper bound, however adding more keywords into the product is still beneficial in these cases.

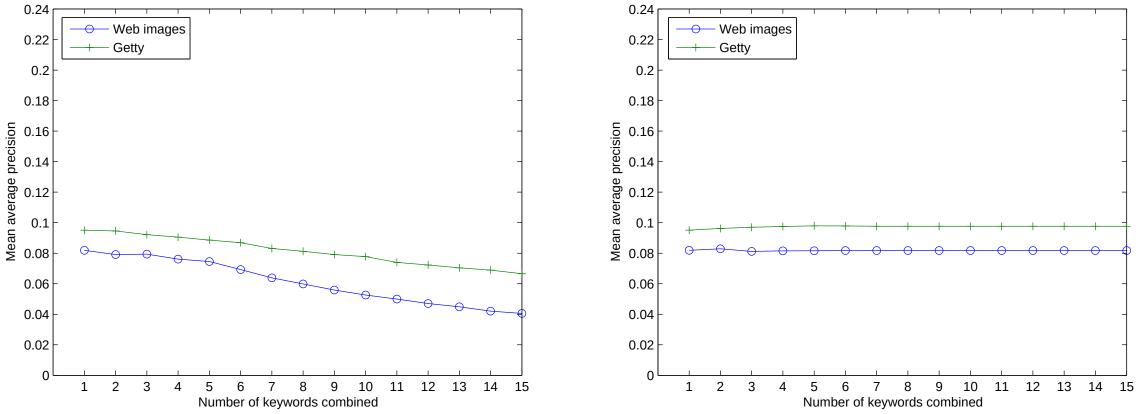
Finally it is worthwhile considering how the same techniques behave when initial keyword probabilities are estimated using much simpler models. Figure 5.3 shows how greedy multiplication and addition perform on both datasets when each Corel keyword is modelled by a single multivariate Gaussian distribution of the keyword’s feature vectors. One can see that in this case greedy addition does not produce an improvement, while greedy multiplication actually degrades performance. This highlights the importance of preserving enough detail of keywords' low-level feature distributions.



(a) Greedy multiplication

(b) Greedy addition

Figure 5.2: Performance of greedy keyword selection vs. the number of combined keywords. Individual keyword models are estimated using nonparametric density estimation.



(a) Greedy multiplication

(b) Greedy addition

Figure 5.3: Performance of greedy keyword selection vs. the number of combined keywords. Individual keyword models are estimated using single Gaussian density functions.

5.3.3 Robustness analysis

Up to this point we have used large quantities of training images to generate relevant keyword combinations. In this section we investigate how the performance of our approach is affected by decreasing the number of positive training examples. For each concept, in turn, we retain a random sample of $n\%$ of all training images relevant to that concept, whilst keeping all other training images which are irrelevant to it. This reflects the situation where the positive examples are hard to obtain, whereas negative images

	Dedicated nonparametric models	Single best keyword	10 keyword product	SVM linear combination	Annotation using keyword vectors as visual features
Average	**0.2188**	**0.1355**	**0.1718**	**0.0997**	**0.1644**
Building Exterior	0.2130	0.1760	0.2126	0.1148	0.1952
Cityscape	0.1531	0.1132	0.1670	0.0772	0.1488
Clear Sky∗	0.3122	**0.1208**	**0.1944**	0.1446	0.2180
Cloud	0.2916	0.2541	0.2490	0.1359	0.2187
Dusk	0.1913	0.1220	0.1503	0.0720	0.1281
Field	0.2533	0.1782	0.1920	0.0836	0.1689
Flower	0.1658	0.0998	0.1136	0.0486	0.1041
Fog	0.1436	0.1018	0.1272	0.0723	0.1018
Food∗	0.4358	**0.0982**	**0.3310**	0.2418	0.2492
Grass	0.2215	0.1977	0.1926	0.1382	0.1685
Horizon	0.1670	0.1063	0.1282	0.0790	0.1308
Landscape	0.1448	0.1070	0.1143	0.0747	0.1245
Leaf	0.2110	0.1159	0.1140	0.1216	0.1655
Lush Foliage	0.2596	0.1303	0.1886	0.1040	0.1701
Mammal∗	0.3227	**0.0906**	**0.2020**	0.1035	0.1994
Mountain	0.1970	0.1332	0.1591	0.0766	0.1663
Night	0.3406	0.2985	0.3189	0.2971	0.2566
Non-Urban Scene	0.1741	0.1269	0.1382	0.0722	0.1467
One Animal∗	0.3482	**0.2346**	**0.3093**	0.1824	0.2715
One Person	0.3289	0.1813	0.2177	0.1206	0.2418
Plant	0.1764	0.1101	0.1252	0.0723	0.1387
River∗	0.1753	**0.0487**	**0.1042**	0.0515	0.1473
Sea	0.1861	0.1470	0.1755	0.0690	0.1649
Sky	0.2804	0.1958	0.2175	0.1453	0.2172
Skyline	0.1709	0.1149	0.1495	0.0992	0.1836
Skyscraper	0.1391	0.0966	0.1435	0.0465	0.1494
Snow	0.1861	0.0996	0.1316	0.0780	0.1315
Sun	0.1510	0.0973	0.0952	0.0445	0.0852
Sunset	0.1988	0.1547	0.1585	0.0854	0.1392
Tree	0.2165	0.1757	0.2173	0.0945	0.1894
Underwater	0.2451	0.1538	0.1776	0.1621	0.1410
Urban Scene	0.1912	0.1766	0.2078	0.0774	0.1818
Vegetable∗	0.1777	**0.0307**	**0.1189**	0.0356	0.1149
Window	0.1772	0.1043	0.1431	0.0621	0.1205
Winter	0.1628	0.0600	0.0915	0.0548	0.1153
Woods	0.1686	0.1250	0.1086	0.0505	0.1227

Table 5.3: Detailed Getty greedy multiplication results. Accuracy of keywords highlighted in bold improves significantly through Corel keyword multiplication.

	Dedicated nonparametric models	Single best keyword	10 keyword product	SVM linear combination	Annotation using keyword vectors as visual features
Average	**0.3341**	**0.0892**	**0.2127**	**0.1783**	**0.2344**
Aerial view	0.3433	0.0686	0.1902	0.1763	0.2414
Building exterior	0.2553	0.0760	0.1529	0.1684	0.2050
Clouds	0.3894	0.2293	0.3105	0.2650	0.2920
Crowd	0.1874	0.0127	0.0581	0.0634	0.0761
Flower	0.2023	0.0602	0.2258	0.1989	0.2320
Grazing animal	0.2632	0.0633	0.1581	0.1822	0.1981
Jug	0.3636	0.0541	0.1043	0.0941	0.1298
Mountain	0.3735	0.1508	0.2184	0.1740	0.3286
Mugshot	0.4449	0.0757	0.2106	0.1848	0.2628
Sunset	0.6150	0.1615	0.5292	0.3604	0.4768
Underwater fish	0.2373	0.0290	0.1814	0.0935	0.1353

Table 5.4: Detailed Web image greedy multiplication results

Web image keyword	Corel keywords selected by the greedy multiplication algorithm
Aerial view	panorama snake hills aerial rocks water nature river stone village
Building exterior	castle building ruins water architecture structure sand river exterior stone
Clouds	cloudy ice clouds wall sky detail beach aerial fortress island
Crowd	pebbles canyon roof bridge wings reflection mammal church river detail
Flower	orchid flora birds mammal leaves rock bird blossoms fortress closeup
Grazing animal	grass wildlife castle field ruins river nature sky animal temple
Jug	furniture sand monument glass hillside fog stone food panorama art
Mountain	mountains structure mountain sky rocks vegetation landscape statue village rock
Mugshot	face waterfall monument beach office scenery texture cliff people rocks
Sunset	sun mountains dusk temple horizon sky scenic sunset street dawn
Underwater fish	underwater mount waterfall fortress reflection fish valley ground scenic birds

Table 5.5: Results of greedy keyword multiplication on Web images. Keywords are shown in the order selected by the algorithm

Getty keyword	Corel keywords selected by the greedy multiplication algorithm
Building Exterior	architecture buildings building rock structure detail sky house park nature
Cityscape	buildings aerial castle city village house detail landscape snow structure
Clear Sky	tower mount statue hill sky rock detail beach ruins city
Cloud	clouds sky park detail landscape bridge cloudy fortress horizon valley
Dusk	sunset buildings clouds rock sky tree horizon park scenic dusk
Field	field mountain vegetation hills cactus landscape grass sky land village
Flower	flora cactus canyon plants lion flower palm nature rock agriculture
Fog	fog panorama waterfall tower detail ground landscape water scenic nature
Food	pets duck meal scenery ground sand castle lion clouds village
Grass	grass building field ruins birds clouds agriculture animal horse vegetation
Horizon	landscape scenic highway sand horizon sky wildlife clouds desert bridge
Landscape	mountain valley vegetation hills hill floor clouds rocks sky park
Leaf	flora vegetation pattern mushroom leaves mountain nature mountains plant palm
Lush Foliage	forest grass reflection vegetation scenery nature horse plants city park
Mammal	frost rocks village lion sun reptile island stone clouds wildlife
Mountain	mountains mount valley castle mountain forest rocks clouds bridge desert
Night	night city scenic vegetation detail architecture evening hillside wings temple
Non-Urban Scene	forest grass reflection mountain nature detail vegetation palm park horse
One Animal	wildlife bird ground animal mountain reptile wings mammal desert nature
One Person	people mammal water detail structure bear person sea building cat
Plant	vegetation nature village blossoms landscape wall closeup mountain plants wildlife
River	bridge river buildings ice mount remains lake lion sun village
Sea	water landscape beach mountains village ocean rocks snow island nature
Sky	sky clouds buildings detail bird hill snow mountain tower cloudy
Skyline	buildings water city sky bridge grass tower structure reflection aerial
Skyscraper	buildings exterior bridge tower detail nature structure landscape street tree
Snow	snow stone mountain island cat remains glacier sport plate winter
Sun	sun scenic sky sunset clouds twilight vegetation silhouette dusk water
Sunset	sun twilight bridge hills sunset clouds landscape sky dusk silhouette
Tree	forest park village vegetation palm garden island detail fortress waterfall
Underwater	underwater glacier nature peak birds fish land harbor tree mountain
Urban Scene	buildings bridge detail city exterior river canyon fountain clock scenic
Vegetable	duck fruit canal tundra pebbles horizon monument sand tower nature
Window	door scenery sunlight winter doorway factory river window cliff detail
Winter	marble horse snow hill rock ice woods river sand remains
Woods	forest reflection grass vegetation detail landscape nature street river summer

Table 5.6: Results of greedy keyword multiplication on Getty images. Keywords are shown in the order selected by the algorithm

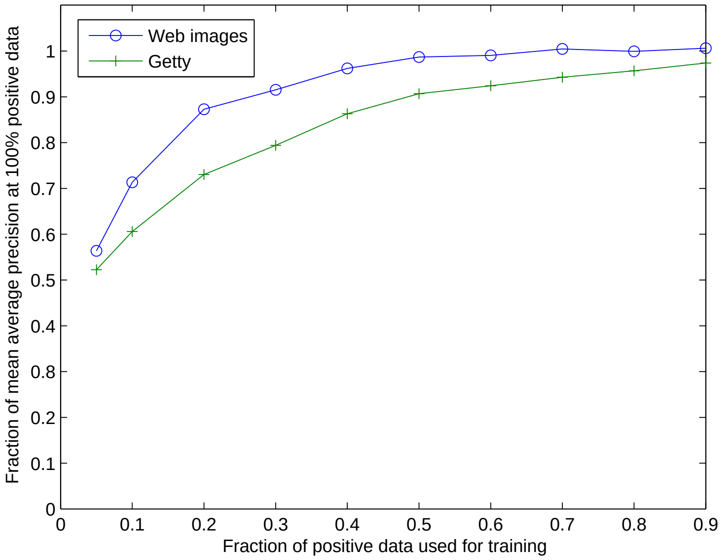


Figure 5.4: Relationship of greedy multiplication performance to the number of positive training examples

are readily available in large quantities. We compute the mean average precision across all concepts for each percentage level n, and to minimise the effect of sampling errors we repeat the entire procedure 5 times and report the average across all trials.

Figure 5.4 reports accuracy as the fraction of mean average precision when 100% of positive examples are used versus the fraction of positive examples retained for each concept. We have chosen this visualisation to make the degradation behaviour comparable across all datasets. The results indicate that the mean average precision degrades gracefully as fewer positive examples remain available, and that performance decreases in a similar manner for both datasets; this shows that our approach is robust towards different selections of positive training examples and towards their reduced availability.

5.4 Conclusions

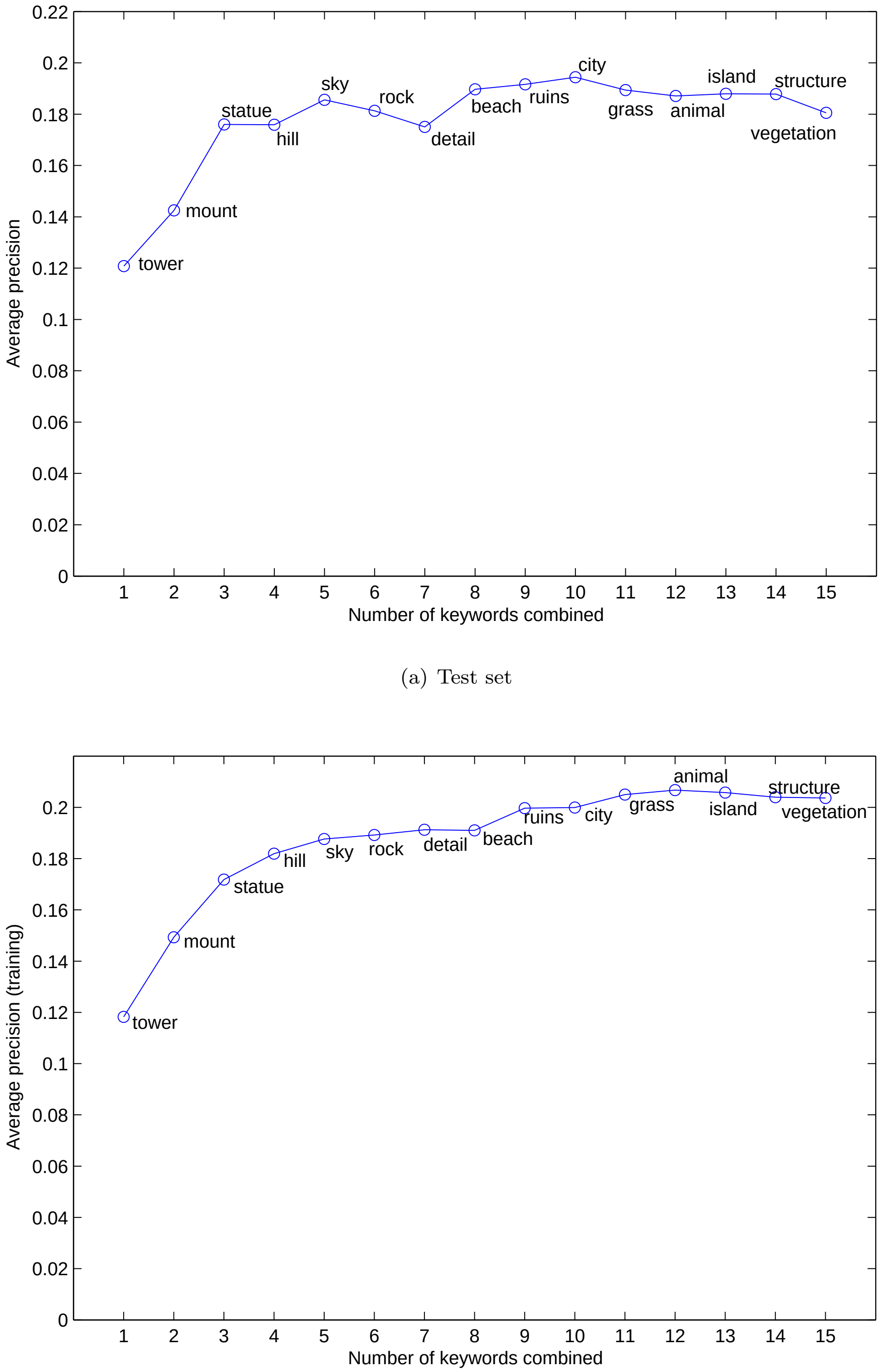
We have presented a framework for efficient re-indexing of large image collections using keyword combination. We have shown that using this simple approach one can refine existing concepts and add new ones into the training vocabulary at a very small computational cost and only a moderate performance tradeoff, compared to a dedicated annotation model. We believe that this functionality could be useful for large scale image search engines when computational resources that are immediately available for re-indexing are scarce.

Keyword multiplication model attains comparable accuracy to the quasi-upper bound for concepts in the Getty and Web datasets. Mean average precision figures for both datasets improve consistently when more keywords are added and degrade similarly when the number of positive examples is reduced. This emphasises the generality of our approach.

It is interesting to observe that the SVM linear combination model is outperformed by simple keyword multiplication. This could be related to the noisy nature of the keyword probabilities computed using the Corel training set. It is possible that the greedy search approach turns out to be more robust in this case. The simple nature of our greedy algorithm leaves room for applying other, more interesting and sophisticated, greedy classification methods in the same manner. Algorithms such as Boosting by Freund and Schapire (1997) or sparse hyperplanes via linear programming by Bhattacharyya et al. (2003) would be suitable candidates.

We believe that in general there is further scope for applying text retrieval techniques to automatically annotated images to enhance users' experience with poorly labelled image collections. One intriguing application of our framework is that of blind relevance feedback for improving image recall. In this scenario, a small set of images is retrieved using sparsely available text metadata, and additional images are then retrieved using a combination of keyword probabilities that best describe the initial results. The robustness of our approach towards small numbers of positive examples, and its computational efficiency, could potentially support such application in near real-time.

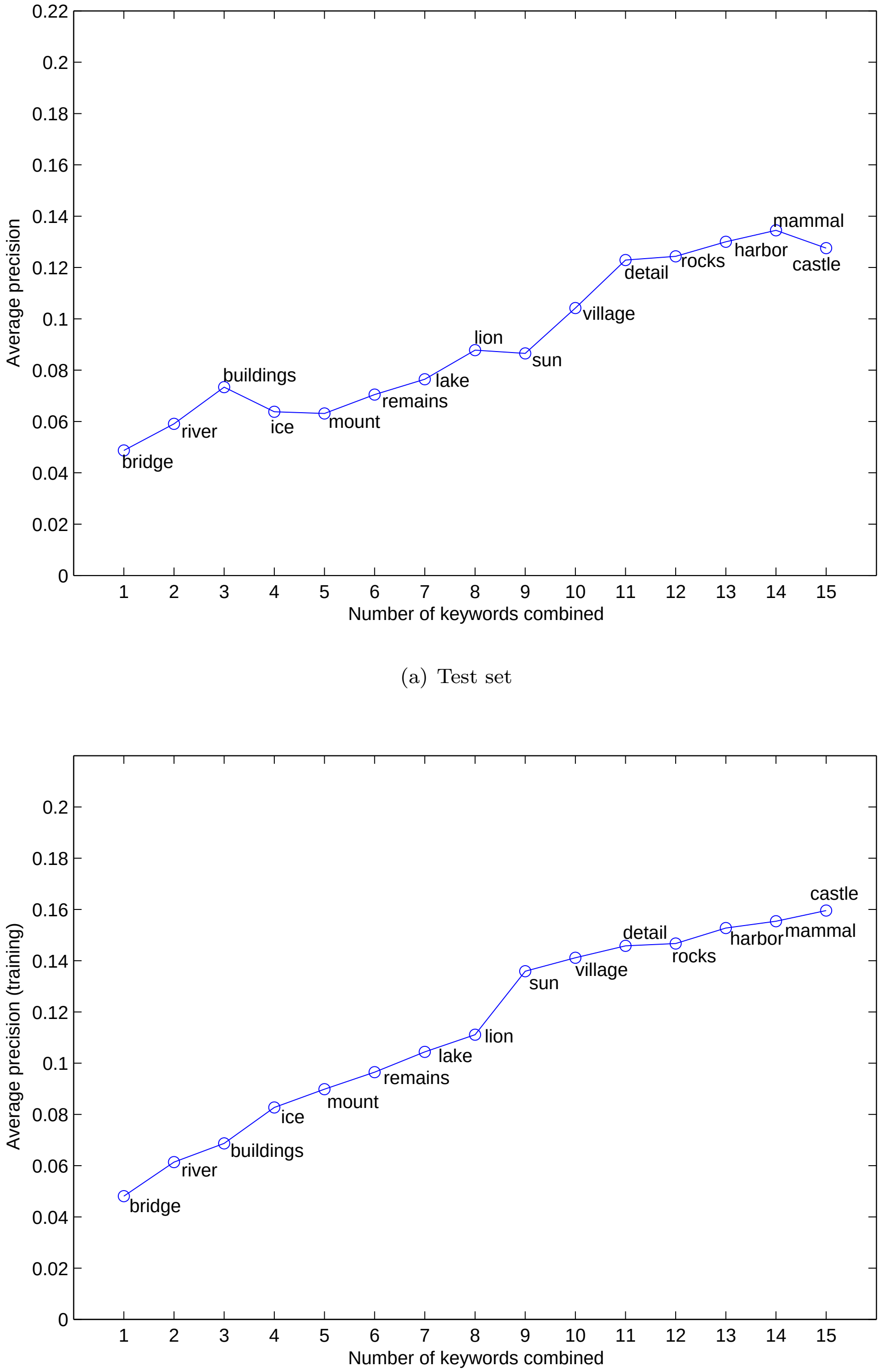
Although we have not applied our re-indexing method to the Behold image search engine, introduced in Chapter 3, we believe that it would be particularly useful in this case. Given the methods inexpensive nature, the search engine could constantly update its annotation models by utilising medium-term user feedback. For example, the act of the user clicking on an image in the search results could be interpreted as a positive relevance judgement for their query. Information gathered in this fashion could be periodically used for quickly re-indexing images with respect to certain concepts.



(a) Test set

(b) Training set

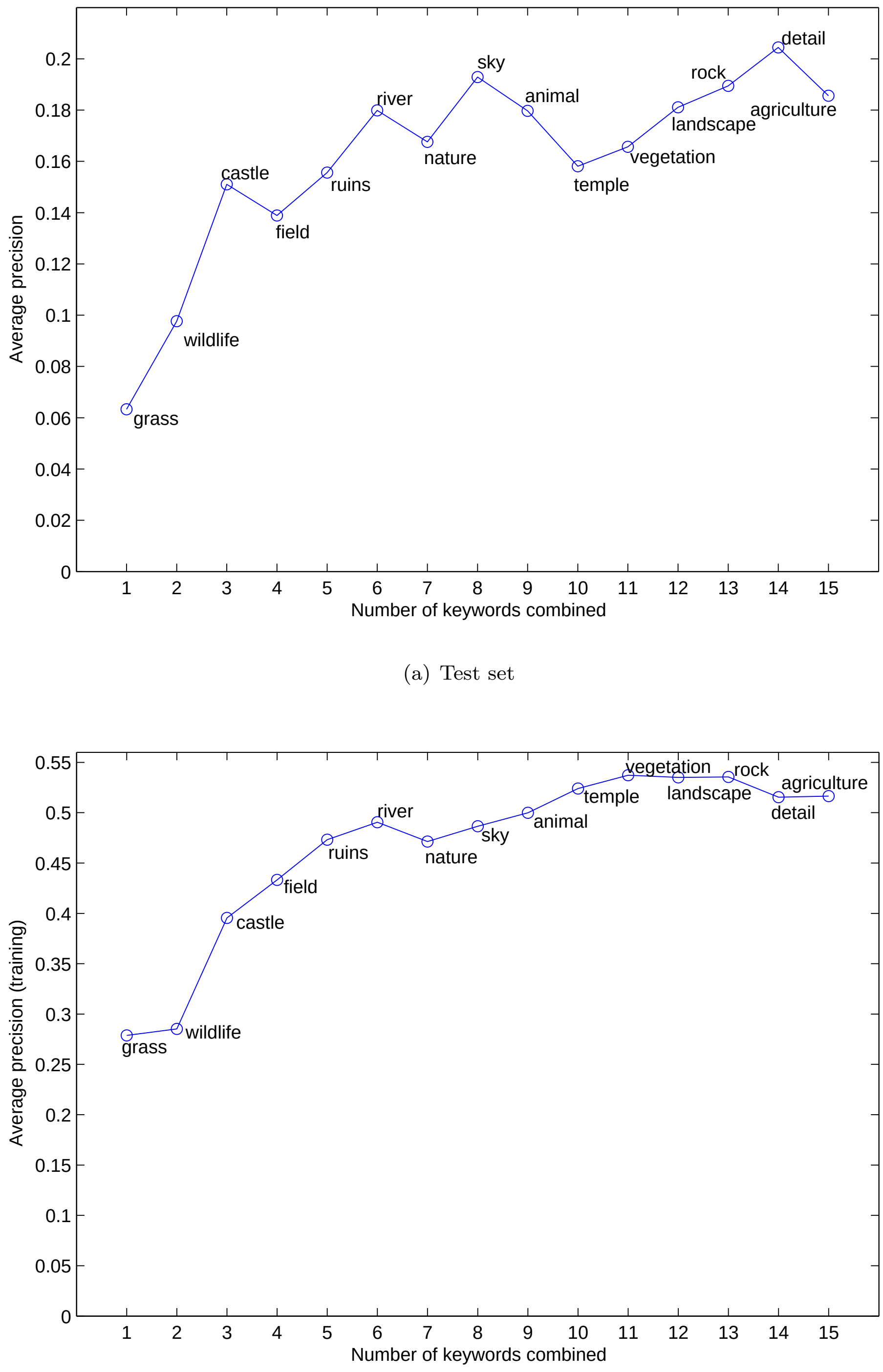
Figure 5.5: Average precision for Getty's *Clear Sky* concept.



(a) Test set

(b) Training set

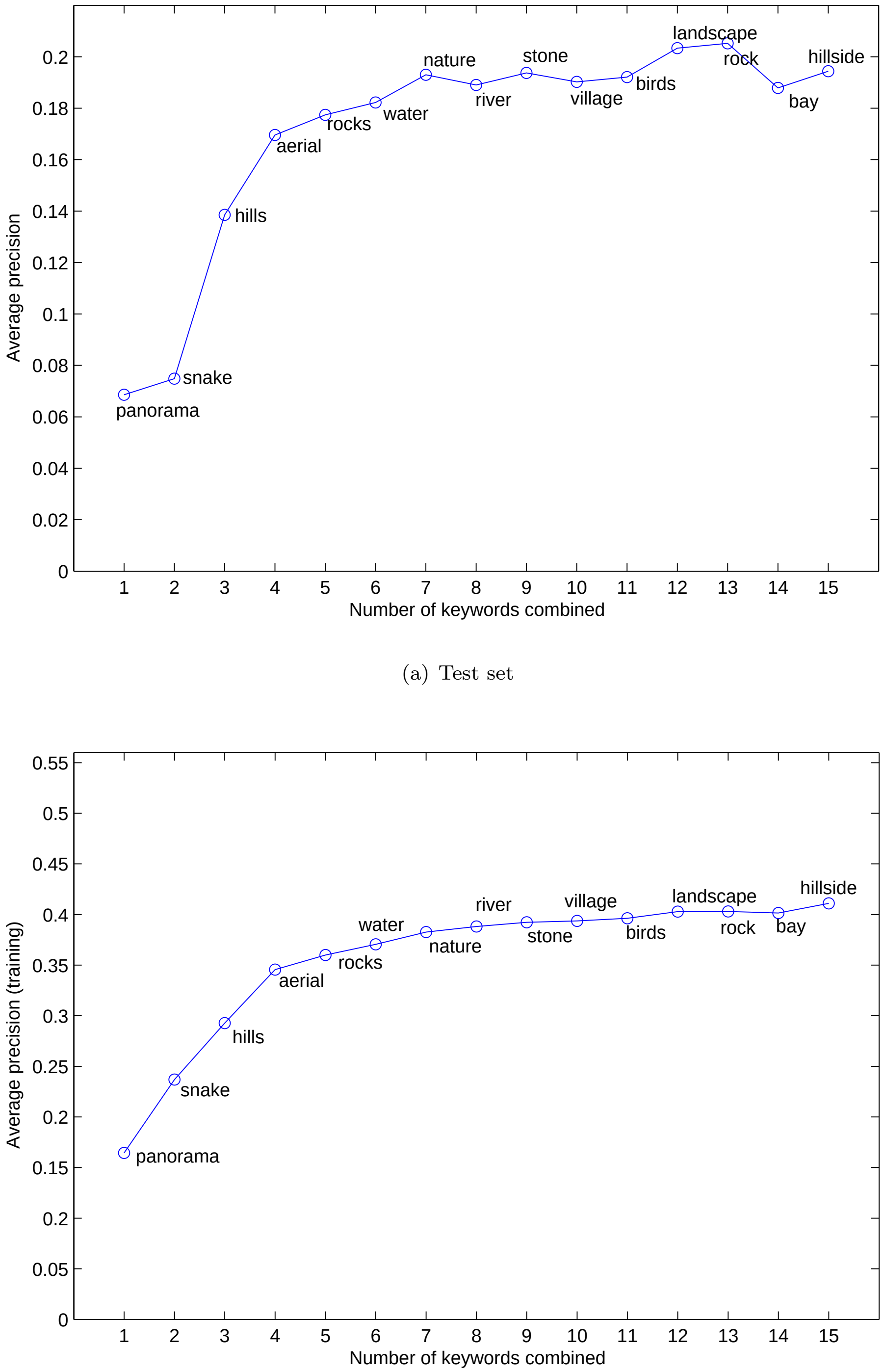
Figure 5.6: Average precision for Getty's *River* concept.



(a) Test set

(b) Training set

Figure 5.7: Average precision for Web image dataset's *Grazing animals* concept.



(a) Test set

(b) Training set

Figure 5.8: Average precision for Web image dataset's *Aerial view* concept.

Chapter 6

Conclusions and Future Work

6.1 Summary of thesis achievements

The original objective of this thesis was to research whether it is possible to index and retrieve photographic images using automated annotation based on simple image properties. For this we modelled manual image annotation as a probabilistic process, in which words are assigned conditionally on global image features. We evaluated this model on one well known benchmark dataset and two new, large image collections. The latter two collections were constructed specifically to reflect realistic retrieval scenarios. We demonstrate that results of state-of-the-art approaches on the benchmark dataset can be rivalled by choosing adequate features. In particular, we show that global colour and texture features are surprisingly well suited for this task. We also show that reasonable retrieval performance for a number of useful image concepts can be achieved on our two new collections. Notably, we demonstrate that — in addition to enabling retrieval of unlabelled images — our image annotation method can be used for improving the accuracy of text-based Internet image search. This can be achieved by re-ranking search results from the metadata index by the probability of appropriate visual concepts. This is a viable practical application of our annotation technique.

Although global features such as colour histograms represent images in a compact manner, they can be high-dimensional. Such high dimensionality may prevent effective probability density estimation for an image annotation model such as ours. In this thesis we propose a technique for modelling probability density functions of histogram data in a manner that is more robust than traditional techniques. We achieve this by performing nonparametric density estimation in the metric space induced by the Earth Mover's Distance. The robust nature of this technique is reflected through superior annotation accuracy.

When we estimate the probability density function for a particular keyword statistically, we risk inadvertently modelling artefacts peculiar to the training images in our collection instead of extracting useful generic patterns. This can result in suboptimal annotation accuracy on unlabelled images. We encountered this effect when we annotated one image collection using keyword models estimated on training images from a different collection. However, through systematic evaluation we find that acceptable annotation accuracy is still maintained in such situations. Likewise, we find that feature distributions of similar keywords in different datasets are indeed similar. This is an encouraging finding that further bolsters the case for our choice of image features for automating image annotation. It shows that while our annotation method is not immune to the aforementioned artefact problem, it is capable of modelling real-world image patterns.

Finally, we find that it is possible to efficiently re-index images that have already been automatically annotated by treating their annotation probabilities as image features. Such re-indexing can be used to improve annotation accuracy or to augment the annotation vocabulary. Images can be re-indexed this way at low computational cost and moderate performance tradeoff compared to repeating the annotation process from scratch.

6.2 Limitations

Besides the general restrictions pointed out by Enser et al. (2005), which we summarised in Section 2.2.1, arguably the two greatest practical limitations of our approach to image annotation are the small vocabulary size and multiple-keyword query interpretation. In this thesis we have considered vocabularies of up to 250 terms. To a lay user this may appear too restrictive for expressing his or her information need. A thesaurus-based tool, such as WordNet (Princeton University, 1998), could ameliorate this problem by mapping the user's diverse query terms onto existing keywords in the vocabulary. This, however, brings us to the challenge of interpreting multi-keyword queries for searching the annotation index. In this thesis we simply take the product of the query keywords' probabilities, yet this may not reflect the user's semantic interpretation of the keyword combination. For example, images of a 'sunny street' do not necessarily have to have the sun visible. To address this problem, Town and Sinclair (2001) have proposed a querying language framework called OQUEL that is based on a context free grammar and a base vocabulary. Words in this language represent predicates on image features and target content at different semantic levels and serve as nouns, adjectives, and prepositions. Sentences are statements of desired characteristics in retrieved images that can represent spatial, object compositional, and more abstract relationships between terms and sub-sentences. Such relationships can be encoded manually to reflect prior knowledge about meanings of keyword combinations. It would be interesting to apply their framework to annotation vocabularies used in this thesis to improve the underlying querying flexibility.

6.3 Future work

A number of new ideas and techniques have been presented in this thesis. Below we outline how some of those lend themselves to further research.

6.3.1 Density estimation in EMD metric space

In this thesis we have proposed the EMD kernel for nonparametric density estimation. It allows one to model densities of *feature distributions* — as opposed to densities of points in a vector space — by representing them as irregularly-quantised histograms. In our experiments this results in superior annotation accuracy compared to when traditional kernel smoothing applied to distributions that are represented by histogram vectors. Yet it is essential that we estimate *proper* densities for this method to be theoretically sound. Proper densities have the property of always integrating to one in the limit. When the kernel has a simple parametric form, such as the Gaussian or the Laplace functions, this property is easy to ensure. As these functions are *translation invariant* in real vector spaces, a single normalisation constant is required. However, it is unclear that this property holds for the EMD kernel. For the EMD kernel density estimate $\hat{f}(x)$ to be a proper density function, the kernel function

$$\int_s e^{-\frac{d(s,s^{(i)})}{h}} \tag{6.1}$$

has to equal some constant C regardless of the basis signature $s^{(i)}$ so that

$$\int_s k_E(s, s^{(i)}; h) ds \tag{6.2}$$

can be normalised to equal one. It is nontrivial to ascertain whether the function (6.1) indeed integrates to a constant because $d(s, s^{(i)})$ is a result of a numerical optimisation procedure for each s. One way to get an indication of whether this property holds in practice could be to use sampling. One might proceed roughly as follows:

Repeat n times:

- Generate a random signature $s^{(i)}$
- Generate m random signatures $g^{(j)}$, $j = 1..m$
- Empirically estimate the integral of $\int_g k_E(g, s^{(i)}; h) dg$ using the above samples g. Denote this integral value $M^{(i)}$

Investigate the variance of values of M. If this variance is small, we can be reasonably confident that the function (6.1) integrates to a constant. However, to get reliable constant estimate both m and n might need to be very large numbers.

6.3.2 Using SIFT features within the EMD density estimation framework

We have observed in Chapter 3 that our way of using the SIFT features within the EMD density estimation framework did not produce satisfactory results. This may be owing to the quantisation method we used to generate the SIFT signatures. Our current approach consists of clustering the SIFT keypoints on a *per-image* basis. In doing so we treated the SIFT descriptors like the wavelet coefficients for the Log-Gabor signature, or the pixel values for the CIE*Lab* signature. In retrospect, it appears that this approach may not take the full advantage of the selective nature of the SIFT keypoints, and may indeed destroy useful information contained therein. The approach taken by Hare et al. (2006) to generate SIFT histograms may be more suitable. They generate a quantisation scheme for SIFT features on a *per-collection* basis by clustering the keypoints from all images in the training set. This quantisation scheme groups similar distinctive keypoints from multiple images and results in a global SIFT *codebook*. The benefit of this approach is that grouped keypoints may potentially relate to image attribtues that are useful for discriminating certain image classes. Histograms that are produced according to this quantisation scheme can still be used within the EMD density estimation framework, as the ground distances between individual histogram bins is known. A natural extension to this approach is to generate a SIFT codebook for each image annotation class independently and then agglomerating these codebooks into the final codebook that is used for generating SIFT histograms. This may retain further helpful, discriminative information extracted by SIFT for each image class.

6.3.3 Identifying the demarcation line between object recognition and image appearance recognition

In this thesis we have used global image features for modelling image probability densities. By doing so we have explicitly assumed that overall image appearance is often indicative of object and scene presence in images. We made this assumption based on the hypothesis put forward by Oliva and Schyns (2000) and Torralba and Oliva (2003). However, this annotation approach is bound to work less effectively for objects that appear within variable settings in images. It would be interesting to determine the extent to which global image appearance is helpful for annotating different image classes. One could go about this by systematically evaluating the discriminative power of global image features for each image class. Object categories that fail to be detected by our annotation method could become the focus of further object recognition research.

6.3.4 Refining Internet image search results through automated annotation

In Section 3.3 we have described our Internet image search engine that allows users to refine their metadata-based queries using automated image annotation. However, currently users have to split their query into two different sub-queries for the respective modalities manually, as is illustrated by the ‘london building’ example, shown in Figure 3.6. It would be easier for an inexperienced user if their query would be split into sub-queries automatically based on the query terms. Intuitively this separation should be based on the expected retrieval accuracy of each term within each modality. A visualness measure similar to the one we proposed in Chapter 4 could be used for making this decision automatically.

Overall, automated image annotation using global image features appears to be — by all means — a useful and a promising approach to image indexing and retrieval. The relative simplicity of its feature extraction and feature modelling steps allows us to apply it to large collections. This, in turn, enables us to investigate exciting new research problems associated with realistic image retrieval scenarios.

Bibliography

S Agarwal, A Awan, and D Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, November 2004.

L Ahn and L Dabbish. Labeling images with a computer game. In *Proceedings of the ACM SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.

A Amir, W Hsu, G Iyengar, C-Y Lin, M Naphade, A Natsev, C Neti, H Nock, J Smith, B Tseng, Y Wu, and D Zhang. IBM research TRECVID-2003 video retrieval system. In *Proceedings of TRECVID*, 2003.

L Baker and A McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Infrmation Retrieval*, 1998.

C Bhattacharyya, L Grate, A Rizki, D Radisky, F Molina, M Jordan, M Bissell, and I Mian. Simultaneous classification and relevant feature identification in high-dimensional spaces: application to molecular profiling data. *Signal Processing*, 83(4):729–743, 2003.

C Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996.

D Blei and M Jordan. Modeling annotated data. In *Proceedings of the ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 127–134, 2003.

C Buckley, G Salton, J Allan, and A Singhal. Automatic query expansion using SMART: TREC 3. In *Text REtrieval Conference*, 1994.

G Carneiro and N Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 163–168, 2005.

O Chapelle, P Haffner, and V Vapnik. SVMs for histogram-based image classification. *IEEE Transactions on Neural Networks, special issue on Support Vector Machines*, 10:1055–1064, 1999.

T Cover and J Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

G Csurka, C Dance, L Fan, J Willamowski, and C Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

S Deerwester, S Dumais, T Landauer, G Furnas, and R Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

P Duygulu, K Barnard, N de Fretias, and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision*, pages 97–112, 2002.

P Enser, C Sandom, and P Lewis. Automatic annotation of images from the practitioner perspective. In *Proceedings of the International Conference in Image and Video Retrieval*, pages 497–506, 2005.

C Faloutsos, R Barber, M Flickner, J Hafner, W Niblack, D Petkovic, and W Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

J Fan, Y Gao, and H Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *Proceedings of the 12th Annual International Conference on Multimedia*, pages 540–547, 2004.

S Feng, R Manmatha, and V Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1002–1009, 2004.

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

D Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

M Flickner, H Sawhney, W Niblack, J Ashley, Q Huang, B Dom, M Gorkahni, J Hafner, D Lee, D Petkovic, D Steele, and P Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28:23–32, September 1995.

Y Freund and R Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

J Friedman, W Stuetzle, and A Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

Y Gao, J Fan, X Xue, and R Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. In *Proceedings of the 14th Annual International Conference on Multimedia*, pages 901–910, 2006.

A Ghoshal, P Ircing, and S Khudanpur. Hidden Markov models for automatic annotation and content based retrieval of images and video. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 544–551, 2005.

M Girolami and C He. Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1253–1264, 2003.

V Govindaraju, S Srihari, and D Sher. A computational model for face location. In *Proceedings of the Third International Conference on Computer Vision*, pages 718–721, 1990.

W Härdle. *Applied Nonparametric Regression.* Cambridge University Press, 1992.

J Hare, P Lewis, P Enser, and C Sandom. A linear-algebraic technique with an application in semantic image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 31–40, 2006.

A Hauptmann, M-Y Chen, M Christel, C Huang, W-H Lin, T Ng, N Papernick, A Velivelli, J Yang, R Yan, H Yang, and H Wactlar. Confounded expectations: Informedia at trecvid 2004. In *Proceedings TRECVID*, 2004.

D Heesch. *The NN^k technique for image searching and browsing.* PhD thesis, Imperial College London, 2005.

D Heesch, A Yavlinsky, and S Rüger. NN^k networks and automated annotation for browsing large image collections from the world wide web. In *Proceedings of the 14th Annual International Conference on Multimedia*, pages 240–244, 2006.

P Howarth and S Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 326–334, 2004.

G Iyengar, P Duygulu, S Feng, P Ircing, S Khudanpur, D Klakow, M Krause, R Manmatha, H Nock, D Petkova, B Pytlik, and P Virga. Joint visual-text modeling for automatic retrieval of multimedia documents. In *Proceedings of the 13th Annual International Conference on Multimedia*, pages 21–30, 2005.

C Jacobs, A Finkelstein, and D Salesin. Fast multiresolution image querying. In *ACM SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286, 1995.

J Jeon and R Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 24–32, 2004.

J Jeon, V Lavrenko, and R Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Infrmation Retrieval*, pages 119–126, 2003.

R Jin, J Chai, and L Si. Effective automatic image annotation via a coherent language model and active learning. In *Proceedings of the 12th Annual International Conference on Multimedia*, pages 892–899, 2004.

T Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142, 1998.

T Joachims. SVM^{light}. http://svmlight.joachims.org/, 2001.

M Jones, J Marron, and S Sheather. A brief survey of bandwidth selection for density estimation. *Journal of American Statistics Association*, 91:401–407, 1996.

T Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces.* PhD thesis, Kyoto University, 1973.

F Kang, R Jin, and J Chai. Regularizing translation models for better automatic image annotation. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 350–359, 2004.

P Kovesi. What are Log-Gabor filters and why are they good, 2003. Available: http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/PhaseCongruency/Docs/convexpl.html.

J Kruskal and M Wish. *Multidimensional Scaling,.* Beverly Hills and London: Sage Publications, 1978.

T Landauer and M Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31–38, Waterloo, Ontario, October 1990.

V Lavrenko, R Manmatha, and J Jeon. A model for learning the semantics of pictures. In *Proceedings of the 16th Conference on Advances in Neural Information Processing Systems NIPS*, 2003.

V Lavrenko, S Feng, and R Manmatha. Statistical models for automatic video annotation and retrieval. In *Proceedings of the IEEE ICASSP International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 17–21, 2004.

E Levina and P Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 251–256, 2001.

R Loader. Bandwidth selection: classical or plug-in? *The Annals of Statistics*, 27(2):415–438, 1999.

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

J Magalhães and S Rüger. Information-theoretic semantic multimedia indexing. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

B Manjunath. *Introduction to MPEG-7, Multimedia Content Description Interface.* John Wiley and Sons, Ltd., 2002.

B Manjunath and W-Y Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

C Manning, P Raghavan, and H Shütze. *Introduction to Information Retrieval.* Cambridge University Press, 2007.

D Metzler and R Manmatha. An inference network approach to image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 42–50, 2004.

M Mitra, A Singhal, and C Buckley. Improving automatic query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.

P Mitra, C Murthy, and S Pal. Density-based multiscale data condensation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24(6):734–747, 2002.

Y Mori, H Takahashi, and R Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

E Mrowka, A Dorado, W Pedrycz, and E Izquierdo. Dimensionality reduction for content-based image classification. In *Proceedings of the Eighth International Conference on Information Visualisation*, pages 435–438, 2004.

Henning Müller, Stéphane Marchand-Maillet, and Thierry Pun. The truth about Corel – evaluation in image retrieval. In *Proceedings of CIVR*, July 2002.

M Naphade and T Huang. A probabilistic framework for semantic indexing and retrieval in video. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 475–478, 2000.

M Naphade, T Kristjansson, and T Huang. Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 536–540, 1998.

A Natsev, M Naphade, and J Smith. Exploring semantic dependencies for scalable concept detection. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 625–628, 2003.

S Nene, K Nayar, and H Murase. Columbia Object Image Library. Technical Report CUCS-006-96, Columbia University, 1996.

W Ng, A Dorado, D Yeung, W Pedrycz, and E Izquierdo. Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error. *Pattern Recognition*, 40(1):19–32, 2007.

A Oliva and P Schyns. Diagnostic colors mediate scene recognition. *Cognitive Psychology*, 41(2):176–210, 2000.

C Papageorgiou, M Oren, and T Poggio. A general framework for object detection. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 555–562, 1998.

E Parzen. On estimation of a probability density and mode. *Annals of Mathematical Statistics*, 35: 1065–1076, 1962.

J Ponce, T Berg, M Everingham, D Forsyth, M Herbert, S Lezebnik, M Marszalek, C Schmid, B Russell, A Torralba, C Williams, J Zhang, and A Zisserman. Dataset issues in object recognition. In *Toward Category-Level Object Recognition*, 2006.

W Press, B Flannery, S Teukolsky, and W Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 1986.

Princeton University. WordNet, online lexical database. http://www.cogsci.princeton.edu/∼wn/, 1998.

A Rao, R Srihari, L Zhu, and A Zhang. A method for measuring the complexity of image databases. *IEEE Transactions on Multimedia*, 4(2):160–173, 2002.

N Rasiwasia, N Vasconcelos, and P Moreno. Query by semantic example. In *Proceedings of the International Conference in Image and Video Retrieval*, pages 51–60, 2006.

R Reiss. Nonparametric estimation of smooth distribution functions. *Scandinavian Journal of Statistics*, 8:116–119, 1981.

B Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, 1996.

S Robertson and K Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

J Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

K Roden. How do people organise their photographs? In *BCS IRSG 21st Annual Colloquium on Information Retrieval Research*, 1999.

Y Rubner. The earth-mover's distance as a metric for image retrieval. Technical Report STAN-CS-TN-98-86, Stanford University, 1998.

Y Rubner, J Puzicha, C Tomasi, and J Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84:25–43, 2001.

G Salton, A Wong, and C Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

E Schofield. *Fitting maximum-entropy models on large sample spaces.* PhD thesis, Department of Computing, Imperial College London, 2006.

F Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

N Sebe, Q Tian, E Loupias, M Lew, and T Huang. Evaluation of salient point techniques. *Image and Vision Computing*, 21:1087–1095, 2003.

J Sivic and A Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

A Smeulders, M Worring, S Santini, A Gupta, and R Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

J Smith and S-F Chang. VisualSEEk: a fully automated content-based image query system. In *ACM Multimedia*, November 1996.

J Smith, M Naphade, and A Natsev. Multimedia semantic indexing using model vectors. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 445–448, 2003.

C Snoek, M Worring, J Geusebroek, D Koelma, and F Seinstra. The mediamill trecvid 2004 semantic video search engine. In *Proceedings TRECVID*, 2004.

M J Swain and D H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

H Tamura. Texture features corresponding to visual perception. *IEEE Transactions. Systems, Man and Cybernetics*, 8(6):460–473, 1978.

J Tang and P Lewis. An image based feature space and mapping for linking regions and words. In *Proceedings of 2nd International Conference on Computer Vision Theory and Applications*, pages 29–35, 2007.

A Torralba and A Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14:391–412, 2003.

C Town and D Sinclair. Ontological query language for content-based image retrieval. Technical Report 2001.1, AT&T Laboratories Cambridge, 2001.

M Turner. Texture discrimination by Gabor functions. *Biological Cybernetics*, 55(2-3):71–82, 1986.

University of Washington. Ground truth image database, 2004. Available: http://www.cs.washington.edu/research/imagedatabase/groundtruth/.

R Vaillant, C Monrocq, and Y Le Cun. Original approach for the localisation of objects in images. *IEEE Proceedings on Visual Image Signal Processing*, 141(4):245–250, 1994.

V Vapnik. *The Nature of Statistical Learning Theory*. SpringerVerlag, 1995.

P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. In *International Conference on Pattern Recognition*, pages 511–518, 2001.

P Viola, N Schraudolph, and T Sejnowski. Empirical entropy manipulation for real-world problems. In *Advances in Neural Information Processing Systems*, volume 8, pages 851–857, 1996.

J Wang and J Li. Learning-based linguistic indexing of pictures with 2-D MHMMs. In *Proceedings of the 10th Annual International Conference on Multimedia*, pages 436–445, 2002.

Y Wu, B Tseng, and J Smith. Ontology-based multi-classification learning for video concept detection. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1003–1006, 2004.

R Yan, M Chen, and A Hauptmann. Mining relationship between video concepts using probabilistic graphical model. In *To appear in the proceedings of IEEE International Conference on Multimedia and Expo*, 2006.

K Yanai and K Barnard. Image region entropy: a measure of ”visualness” of web images associated with one concept. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 419–422, 2005.

A Yavlinsky. Behold image search engine, 2005. Online at: http://www.beholdsearch.com/.

A Yavlinsky and S Rüger. Efficient re-indexing of automatically annotated image collections using keyword combination. In *Proceedings of SPIE Volume 6506. Multimedia Content Access: Algorithms and Systems*, 2007.

A Yavlinsky, E Schofield, and S Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 507–517, 2005.

A Yavlinsky, D Heesch, and S Rüger. A large scale system for searching and browsing images from the world wide web. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 530–534, 2006.

Appendix A

Additional Figures for Chapter 4

Building Exterior
Urban Scene, *House*, Tree, *City*, *Structure*, *Cityscape*, Rock, Non-Urban Scene, *Town*(`Joint feature`)
Urban Scene, *House*, *City*, Rock, *Structure*, Tree, *Cityscape*, Sky, Cloud (`MargCIE-3×3`)
Urban Scene, *House*, *Structure*, Tree, *Cityscape*, Non-Urban Scene, *City*, River, Rock (`Tamura-3×3`)
Urban Scene, *City*, *House*, *Structure*, Tree, *Cityscape*, *Church*, *Dome*, Old Ruin (`Log-Gabor`)
Cityscape
Building Exterior, *Urban Scene*, *City*, *Town*, Tree, Structure, House, Harbor, Rock
Urban Scene, *City*, *Building Exterior*, *Skyscraper*, *Town*, Mountain, Sky, Harbor, Structure
Building Exterior, *Urban Scene*, *Structure*, Tree, *Town*, River, Non-Urban Scene, Rock, *City*
Tree, House, *Town*, *Building Exterior*, *City*, *Urban Scene*, *Structure*, Roof, Harbor
Clear Sky
Sky, Cloud, Mountain, Beach, Sea, Snow, Sunlight, Hill, Water
Sky, Cloud, Building Exterior, Mountain, Sea, Horizon, Structure, City, Hill
Sky, Sea, Mountain, Beach, Snow, Horizon, Hill, Water, Building Exterior
Night, Snow, *Sky*, Tower, Winter, Sunlight, One Animal, Water, Cliff
Cloud
Sky, Mountain, Horizon, Sea, Beach, Landscape, Hill, Coastline, Lake
Sky, Horizon, Mountain, Sea, City, Landscape, Hill, Lake, Structure
Sky, Sea, Mountain, Water, Sunlight, Beach, Horizon, Landscape, Lake
Mountain, Horizon, Beach, Landscape, Coastline, Sea, *Sky*, Dusk, Lake
Dusk
Cloud, Sky, Reflection, Mountain, Sea, *Night*, *Twilight*, Water, Silhouette
Urban Scene, Sunset, Cityscape, Silhouette, Building Exterior, *Twilight*, *Night*, Skyscraper, Reflection
Cloud, Sky, Sea, Horizon, Mountain, Beach, Lake, Landscape, Water
Cloud, Sky, Beach, Sea, Lake, Mountain, Coastline, Water, Reflection
Field
Rural Scene, *Grass*, *Non-Urban Scene*, *Farm*, Hill, Landscape, *Crop*, Tree, Mountain
Rural Scene, *Grass*, *Crop*, *Farm*, Tree, *Non-Urban Scene*, Plant, Hill, Sunlight
Rural Scene, *Grass*, *Non-Urban Scene*, *Farm*, *Crop*, Plant, Tree, Landscape, Hill
Rural Scene, Hill, Extreme Terrain, *Non-Urban Scene*, Mammal, Mountain, *Farm*, *Grass*, Snow
Flower
Plant, Tree, Summer, Non-Urban Scene, One Animal, Rural Scene, Animals In The Wild, Building Exterior, *Leaf*
Plant, Non-Urban Scene, Tree, Crop, Field, Rural Scene, Summer, Animals In The Wild, *Leaf*
Plant, Non-Urban Scene, Tree, Summer, Animals In The Wild, One Animal, Rural Scene, Autumn, *Leaf*
Plant, Tree, Summer, Animals In The Wild, Branch, *Wildflower*, Leaf, Non-Urban Scene, Urban Scene
Fog
Cloud, Surf, Horizon, Mountain, *Overcast*, Landscape, Sand, Dawn, Beach
Snow, *Overcast*, Winter, Surf, Ice, Mountain, *Cloud*, Horizon, Mammal
Cloud, Mountain, Sea, Water, Sky, Landscape, Sand, Coastline, Beach
Romantic Sky, Sunset, Volcano, Sun, Sunrise, Iceberg, Surf, Dawn, Flat
Food
Vegetable, *Plate*, *Fruit*, One Person, Table, *Dessert*, *Bread*, Women, Men
Vegetable, *Plate*, *Fruit*, One Person, Table, *Bread*, *Dessert*, Chair, Wood
Vegetable, *Fruit*, *Dessert*, *Plate*, *Bread*, *Tomato*, Table, One Person, People
Vegetable, *Plate*, *Bread*, *Fruit*, *Bowl*, *Dessert*, Men, People, *Tomato*
Grass
Rural Scene, Non-Urban Scene, Tree, *Field*, *Plant*, Sunlight, Summer, Rock, One Animal
Rural Scene, *Field*, Non-Urban Scene, Tree, *Lawn*, Summer, Farm, Sunlight, Forest
Rural Scene, Non-Urban Scene, *Field*, Tree, *Plant*, Rock, Building Exterior, Structure, House
Non-Urban Scene, *Rural Scene*, Tree, Rock, *Plant*, Castle, Sunlight, *Field*, Summer
Horizon
Cloud, Mountain, *Landscape*, *Sky*, Sea, Beach, Hill, Coastline, Extreme Terrain
Cloud, *Sky*, Mountain, Beach, Sea, *Landscape*, Hill, Snow, Coastline
Sky, Cloud, Sea, Landscape, Mountain, Beach, Dusk, Hill, Sand
Landscape, Cloud, Mountain, Island, Mountain Range, Overcast, Coastline, Sea, Flat
Landscape
Mountain, Cloud, Hill, *Horizon*, Extreme Terrain, Sky, Lake, Coastline, Sea
Hill, Mountain, Cloud, Rock, Sunlight, Horizon, Sky, Rural Scene, River
Hill, Rural Scene, Mountain, Field, Non-Urban Scene, Grass, Extreme Terrain, Sunlight, Cloud
Mountain, *Horizon*, Mountain Range, Cloud, Rock Formation, Extreme Terrain, Hill, Coastline, Mountain Peak
Leaf
Plant, *Tree*, *Flower*, Woods, Non-Urban Scene, Forest, Autumn, Lush Foliage, Branch
Plant, Woods, Forest, *Tree*, Lush Foliage, *Flower*, Non-Urban Scene, Animals In The Wild, Crop
Plant, Autumn, Tree, Non-Urban Scene, Forest, *Flower*, Woods, Mammal, Branch
Plant, Tree, *Flower*, Autumn, Branch, Spring, Rainforest, *Wildflower*, Forest

Table A.1: Within-dataset keyword distribution similarities for Getty Images, part I

Lush Foliage
Rainforest, *Woods*, *Forest*, Tree, *Plant*, Non-Urban Scene, Grass, Leaf, Rural Scene (`Joint feature`)
Rainforest, *Woods*, *Forest*, Grass, Leaf, Non-Urban Scene, Tree, *Plant*, Summer (`MargCIE-3×3`)
Rainforest, *Woods*, *Forest*, Crop, Tree, Plant, Autumn, Non-Urban Scene, Farm (`Tamura-3×3`)
Rainforest, *Forest*, *Woods*, Tree, *Plant*, Autumn, Non-Urban Scene, Footpath, Rock (`Log-Gabor`)
Mammal
Animal, River, *One Animal* Rock, Snow, *Deer*, Mountain, Non-Urban Scene, Winter
Animal, River, Wolf, Deer, Bird, Rock, *One Animal*, Stone, One Person
Plant, Non-Urban Scene, Tree, Rural Scene, *Animal*, *Animals In The Wild*, *One Animal*, Grass, River
Animal, Rural Scene, Extreme Terrain, Rock Formation, Mountain, Hill, Cliff, Field, Snow
Mountain
Cloud, *Extreme Terrain*, Landscape, *Hill* Sky, Horizon, Coastline, Sea, *Mountain Peak*
Cloud, *Extreme Terrain*, *Hill* Sky, Landscape, Rock, River, Lake, City
Hill, *Extreme Terrain*, Landscape, Rural Scene, Snow, Cloud, Non-Urban Scene, Sunlight, Field
Extreme Terrain, *Mountain Range*, Landscape, *Mountain Peak*, Cloud, *Rock Formation*, Coastline, *Hill*, Horizon
Night
Dusk, One Animal, Urban Scene, Non-Urban Scene, Water, Animals In The Wild, Reflection, Building Exterior, Sky
Dusk, Urban Scene, Traffic, Building Exterior, Cityscape, Tree, *Dark*, Skyscraper, Car
One Animal, Sky, Urban Scene, Water, Building Exterior, Snow, Sea, Animals In The Wild, Non-Urban Scene
Clear Sky, Sky, Water, Snow, One Animal, Winter, Reflection, *Dusk*, Sunlight
Non-Urban Scene
Tree, *Rural Scene*, Sunlight, One Animal, Rock, Plant, *Forest*, Grass, Water
Tree, Sunlight, *Forest*, One Animal, Rural Scene, Plant, Building Exterior, Summer, House
Tree, Plant, Grass, *Rural Scene*, Rock, *Forest*, *Field*, Mammal, Sunlight
Grass, Rock, *Rural Scene* Sunlight, Tree, Plant, Winter, Snow, Summer
One Animal
Animal, *Animals In The Wild*, Non-Urban Scene, *Animal Head*, Sunlight, Water, Rock, Tree, *Mammal*
Animal, Tree, Non-Urban Scene, *Animals In The Wild*, *Animal Head*, Sunlight, *Mammal*, Water, Rock
Animals In The Wild, *Animal*, Water, Underwater, Non-Urban Scene, Sunlight, Plant, *Mammal*, Snow
Animal, *Animals In The Wild*, *Animal Head*, Winter, Underwater, Snow, Sunlight, Night, Water
One Person
Head And Shoulders, Food, One Animal, *People*, *Men*, *Women*, Beach, Bed, Animal
Window, *Men*, *Women*, Beach, Rock, Wall, Animal, Food, Office
Food, One Animal, *Men*, Water, Women, Urban Scene, Table, Animals In The Wild, *People*
Head And Shoulders, *People*, Bed, Bedroom, Human Hand, Drink, Vase, Lamp, Dessert
Plant
Tree, Non-Urban Scene, *Flower*, Forest, Rural Scene, *Grass*, One Animal, Rock, Summer
Tree, Non-Urban Scene, Forest, One Animal, Rural Scene, Woods, *Flower*, Animals In The Wild, Sunlight
Tree, Non-Urban Scene, Forest, Autumn, *Flower*, Rural Scene, *Leaf*, Mammal, *Grass*
Tree, *Flower*, Non-Urban Scene, Rock, *Forest*, *Grass*, Summer, *Leaf*, Rainforest
River
Rock, Mammal, *Water*, Structure, Rural Scene, Non-Urban Scene, Mountain, Hill, Snow
Rock, City, Structure, Mountain, Mammal, Stone, Sunlight, Cloud, Coastline
Non-Urban Scene, Tree, Building Exterior, Urban Scene, Cityscape, Rock, Structure, Plant, House
Rock, Castle, Rural Scene, Nautical Vessel, *Waterfront*, Harbor, Hill, Cliff, Mammal
Sea
Cloud, *Beach*, Sky, Mountain, *Coastline*, Horizon, *Water*, Snow, Lake
Cloud, Sky, *Beach*, *Water*, Horizon, *Coastline*, Mountain, Snow, Lake
Cloud, *Beach*, Sky, *Coastline*, *Water*, Sand, Sunlight, Urban Scene, Mountain
Beach, *Coastline*, Cloud, Mountain, Dusk, Horizon, Lake, Hill, Island
Sky
Cloud, Mountain, Beach, Sea, *Clear Sky*, *Horizon*, Snow, Landscape, Hill
Cloud, *Clear Sky*, *Horizon*, Mountain, Sea, City, Building Exterior, Urban Scene, Lake
Cloud, Sea, *Clear Sky*, Beach, *Horizon*, Water, Mountain, One Animal, Snow
Cloud, Dusk, Mountain, Beach, Mountain Peak, Night, Snow, Sea, Reflection
Skyline
Harbor, Building Exterior, *Cityscape*, *Urban Scene*, *City*, Tower, Structure, *Sky*, *Clear Sky*
Skyscraper, Harbor, *Cityscape*, *Urban Scene*, Building Exterior, *City*, *Sky*, Cloud, Mountain
Harbor, *Cityscape*, Waterfront, Building Exterior, *Sky*, Urban Scene, Tower, Structure, *Clear Sky*
Harbor, Tower, Waterfront, Castle, Structure, Nautical Vessel, Church, *City*, Bridge
Skyscraper
Urban Scene, *Building Exterior*, *City*, *Cityscape*, *Structure*, Tree, Skyline, House, *Tower*
Urban Scene, *Building Exterior*, *Cityscape*, *City*, Reflection, Sky, Cloud, Mountain, Harbor
Building Exterior, *Urban Scene*, *City*, *Cityscape*, *Structure*, River, Non-Urban Scene, Tree, Sunlight
Building Exterior, *Urban Scene*, *City*, *Cityscape*, *Structure*, Church, Dome, *Building Structure*, Palm Tree

Table A.2: Within-dataset keyword distribution similarities for Getty Images, part II

Snow
Winter, Water, Beach, Sky, Mountain, Ice, Cloud, Sea, Rock (`Joint feature`)
Winter, Ice, Beach, Surf, Cloud, Sky, Mountain, Glacier, Sea (`MargCIE-3×3`)
Winter, Non-Urban Scene, Water, One Animal, Urban Scene, Building Exterior, Animals In The Wild, Mountain, Sunlight (`Tamura-3×3`)
Winter, Water, Sky, Beach, Cliff, Clear Sky, Sunlight, One Animal, Hill (`Log-Gabor`)
Sun
Sunset, Silhouette, *Sunrise*, Cloud, Dawn, Twilight, Dusk, Horizon, Landscape
Sunset, Silhouette, *Sunrise*, Dusk, Twilight, Dawn, Urban Scene, Wood, Sunlight
Sunset, Cloud, Horizon, Sky, Dusk, Sea, Beach, Fog, Silhouette
Sunset, Fog, Sunrise, Romantic Sky, Volcano, Dawn, Iceberg, Airplane, Flat
Sunset
Sun, *Dusk*, Silhouette, Cloud, Sunrise, Twilight, Dawn, Horizon, Mountain
Sun, *Dusk*, Silhouette, Sunrise, Twilight, Dawn, Urban Scene, Moody Sky, Wood
Horizon, *Dusk*, Sea, Cloud, Sky, *Sun*, Silhouette, Beach, Sand
Sun, Sunrise, Dawn, Fog, Romantic Sky, Horizon, Flat, Iceberg, Cloud
Tree
Non-Urban Scene, *Plant*, Building Exterior, *Forest*, House, Rock, *Rural Scene*, Urban Scene, Grass
Non-Urban Scene, *Forest*, House, *Plant*, Sunlight, One Animal, Building Exterior, *Rural Scene*, Urban Scene
Plant, Non-Urban Scene, *Forest*, Autumn, *Rural Scene*, Rock, Grass, House, Building Exterior
Plant, *Forest*, Non-Urban Scene, Grass, Cityscape, Rock, Building Exterior, Structure, House
Underwater
One Animal, Animals In The Wild, Water, Animal, Non-Urban Scene, Plant, Sunlight, Rock, Tree
Animals In The Wild, Water, Tree, One Animal, Plant, Forest, Non-Urban Scene, Building Exterior, Urban Scene
One Animal, Animals In The Wild, Animal, Plant, Mammal, Non-Urban Scene, Leaf, Water, Snow
One Animal, Animal, Snow, Animals In The Wild, Water, Winter, Sunlight, Non-Urban Scene, Fish
Urban Scene
Building Exterior, *City*, *Cityscape*, House, Tree, Structure, Rock, Statue, Building Structure
Building Exterior, *City*, *Cityscape*, House, Reflection, *Skyscraper*, Tree, Sky, Cloud
Building Exterior, *City*, *Structure*, *Cityscape*, Non-Urban Scene, House, River, Tree, Rock
Building Exterior, *City*, *Structure*, *Cityscape*, Statue, Church, Dome, Palm Tree, House
Vegetable
Food, Fruit, Plate, Bread, One Person, Table, *Tomato*, One Animal, Women
Food, Fruit, Plate, Bread, Table, One Person, *Tomato*, Chair, Wood
Food, Fruit, *Tomato*, Bread, Plate, Dessert, Flower, Bowl, Table
Food, Plate, Bread, Fruit, *Tomato*, Bowl, Men, People, Dessert
Window
Urban Scene, *Building Exterior*, Table, Chair, City, *House*, One Person, *Building Structure*, Sunlight
One Person, Urban Scene, *Building Exterior*, Rock, City, Reflection, Sunlight, Office, Table
Urban Scene, Chair, Table, *Building Exterior*, Wood, Water, Car, One Person, Reflection
Chair, *Building Exterior*, Urban Scene, Table, Sofa, Statue, Wood, City, *Building Structure*
Winter
Snow, *Ice*, Beach, Water, Rock, Sky, River, One Animal, Mammal
Snow, *Ice*, Beach, One Person, River, Surf, Cloud, Deer, Fog
Snow, Non-Urban Scene, One Animal, Building Exterior, Urban Scene, Water, Tree, Animals In The Wild, Plant
Snow, Water, One Animal, Non-Urban Scene, Sunlight, Night, Animal, Rural Scene, Clear Sky
Woods
Forest, *Tree*, *Lush Foliage*, *Rainforest*, *Plant*, Non-Urban Scene, Leaf, Grass, Autumn
Forest, *Tree*, Non-Urban Scene, *Plant*, *Lush Foliage*, One Animal, Branch, *Rainforest*, Summer
Forest, *Lush Foliage*, *Rainforest*, Autumn, *Tree*, Plant, Crop, Non-Urban Scene, Leaf
Autumn, *Tree*, *Rainforest*, *Lush Foliage*, *Forest*, *Plant*, Footpath, Stone, Wildflower

Table A.3: Within-dataset keyword distribution similarities for Getty Images, part III

animal
mammal, *wildlife*, *nature*, *birds*, water, vegetation, rocks, rock, stone (`Joint feature`)
mammal, *wildlife*, *bird*, *birds*, water, stone, rocks, rock, architecture (`MargCIE-3×3`)
wildlife, *mammal*, *nature*, vegetation, grass, rock, leaves, tree, *birds* (`Tamura-3×3`)
mammal, *wildlife*, *nature*, vegetation, *birds*, plant, rock, grass, *cat* (`Log-Gabor`)
building
architecture, *buildings*, *structure*, exterior, *castle*, ruins, stone, sky, *city*
architecture, *buildings*, sky, water, exterior, *structure*, landscape, scenic, ruins
architecture, *buildings*, *structure*, exterior, water, ruins, sky, vegetation, *castle*
architecture, *castle*, *buildings*, *structure*, exterior, ruins, city, temple, *church*
city
buildings, *architecture*, *building*, sky, *structure*, exterior, water, stone, *bridge*
buildings, sky, *building*, *architecture*, water, scenic, landscape, clouds, exterior
buildings, *building*, *architecture*, water, sky, *structure*, vegetation, rock, exterior
buildings, *building*, *structure*, *architecture*, exterior, castle, vegetation, ruins, tower
clouds
sky, landscape, water, scenic, mountains, mountain, valley, island, scenery
sky, landscape, building, architecture, water, scenic, buildings, mountains, mountain
sky, landscape, water, scenic, mountains, building, architecture, sand, buildings
landscape, *sky*, scenic, water, mountains, island, scenery, hills, valley
dusk
sunset, sun, *evening*, dawn, *twilight*, *nightfall*, clouds, silhouette, sky
sunset, sun, *evening*, *twilight*, *nightfall*, dawn, night, reflection, detail
sunset, sun, dawn, *evening*, sky, beach, horizon, clouds, scenic
sunset, sun, dawn, *evening*, silhouette, cloudy, sunrise, mist, *twilight*
field
grass, landscape, *vegetation*, water, scenic, sky, nature, animal, wildlife
grass, *vegetation*, landscape, animal, tree, water, wildlife, scenic, nature
grass, *vegetation*, nature, *plants*, rock, landscape, park, stone, *agriculture*
grass, landscape, water, scenic, sky, scenery, mountain, *vegetation*, aerial
flower
plant, closeup, nature, *flora*, *leaves*, *vegetation*, fruit, *plants*, wildlife
plant, closeup, *plants*, *leaves*, food, *flora*, insect, fruit, *blossoms*
flora, *plant*, closeup, *leaves*, nature, wildlife, animal, mammal, *vegetation*
plant, closeup, mammal, animal, wildlife, nature, *rose*, wings, birds
fog
mist, landscape, mountains, *clouds*, scenic, water, sky, valley, scenery
mist, mountains, scenery, landscape, buildings, water, structure, castle, architecture
mist, landscape, *clouds*, water, scenic, sky, mountains, valley, mountain
mist, dawn, sunrise, lighthouse, *clouds*, smoke, wilderness, landscape, valley
food
fruit, leaves, plant, closeup, agriculture, nature, flower, plants, detail
fruit, leaves, plant, insect, meal, closeup, plants, flower, agriculture
fruit, pebbles, leaves, detail, plant, closeup, nature, wildlife, animal
fruit, *drink*, leaves, nature, rocks, wildlife, agriculture, flora, plant
grass
vegetation, *field*, landscape, water, wildlife, animal, mammal, sky, stone
field, animal, mammal, wildlife, vegetation, landscape, water, tree, stone
vegetation, *field*, nature, stone, animal, rock, landscape, castle, wildlife
vegetation, water, *field*, landscape, sky, mountain, park, scenic, wildlife
horizon
landscape, clouds, sky, scenic, mountains, water, island, valley, hills
landscape, sky, clouds, scenic, water, architecture, *panorama*, buildings, building
clouds, *landscape*, sky, scenic, water, valley, mountains, island, hills
clouds, *landscape*, hills, island, mountains, scenic, valley, beach, wilderness

Table A.4: Within-dataset keyword distribution similarities for Corel-16k, part I

landscape
water, scenic, sky, clouds, mountains, mountain, valley, scenery, grass (`Joint feature`)
water, sky, scenic, architecture, building, mountains, clouds, buildings, stone (`MargCIE-3×3`)
water, scenic, sky, vegetation, mountains, buildings, valley, clouds, mountain (`MargCIE-3×3`)
clouds, scenic, water, sky, mountains, scenery, valley, island, hills (`Log-Gabor`)
leaf
nature, closeup, *vegetation*, wings, animal, wildlife, *plant*, detail, *leaves*
insect, nature, *leaves*, wings, *plant*, autumn, *vegetation*, *flora*, agriculture
texture, nature, pattern, detail, *flora*, autumn, marble, *leaves*, abstract
flower, closeup, nature, *plant*, birds, land, wildlife, animal, mammal
mammal
animal, *wildlife*, rock, grass, rocks, *cat*, birds, vegetation, stone
animal, *wildlife*, rocks, stone, birds, *cat*, bird, water, architecture
animal, *wildlife*, nature, vegetation, leaves, grass, rock, tree, birds
animal, *wildlife*, *cat*, vegetation, nature, rock, grass, plant, birds
mountain
landscape, water, sky, *mountains*, scenic, clouds, grass, park, castle
landscape, water, sky, *mountains*, building, architecture, clouds, scenic, buildings
landscape, vegetation, grass, *mountains*, water, architecture, building, castle, park
water, landscape, *mountains*, sky, scenic, scenery, clouds, river, grass
nature
vegetation, *wildlife*, *animal*, tree, rock, leaves, detail, *forest*, *mammal*
vegetation, *tree*, detail, *forest*, *animal*, *wildlife*, *birds*, water, ground
vegetation, *leaves*, *animal*, *wildlife*, detail, rock, *tree*, *plants*, grass
vegetation, *wildlife*, *animal*, rock, *mammal*, rocks, park, *birds*, leaves
night
sky, reflection, water, city, scenic, landscape, skyline, *evening*, closeup
evening, *dusk*, fireworks, scene, reflection, city, sunset, closeup, plant
city, water, grass, sky, scene, buildings, field, nature, rock
sky, skyline, landscape, water, reflection, *evening*, scenic, monument, clouds
person
people, animal, mammal, wildlife, water, rock, sky, closeup, stone
people, detail, animal, rock, art, stone, water, wildlife, mammal
people, plant, flower, closeup, animal, nature, fruit, wildlife, mammal
people, *women*, mammal, animal, orchid, flower, *face*, plant, sky
plant
leaves, nature, closeup, *flower*, *vegetation*, wildlife, animal, *tree*, detail
leaves, nature, closeup, *plants*, *vegetation*, insect, *flower*, texture, detail
leaves, *flower*, closeup, nature, *vegetation*, detail, animal, wildlife, *flora*
flower, closeup, animal, mammal, wildlife, nature, *vegetation*, wings, birds
river
water, landscape, scenic, rocks, grass, sky, animal, mountains, wildlife
water, landscape, animal, architecture, scenic, rocks, wildlife, stone, mammal
vegetation, *water*, nature, architecture, rocks, grass, landscape, rock, animal
water, mountain, sky, landscape, scenery, grass, scenic, mountains, vegetation
sea
water, landscape, scenic, sky, clouds, island, scenery, mountains, rock
water, landscape, scenic, sky, architecture, buildings, clouds, scenery, animal
water, landscape, scenic, sky, sand, vegetation, buildings, rock, animal
water, landscape, scenic, clouds, sky, island, mountains, mountain, scenery
skyline
city, *sky*, *buildings*, water, scenic, landscape, clouds, reflection, tower
city, *buildings*, *sky*, water, clouds, *architecture*, *building*, scenic, harbor
city, *buildings*, *sky*, landscape, water, structure, scenery, scenic, *building*
night, *sky*, tower, water, reflection, scenic, landscape, clouds, bridge

Table A.5: Within-dataset keyword distribution similarities for Corel-16k, part II

snow
winter, mammal, water, animal, rocks, architecture, wildlife, building, stone (`Joint feature`)
winter, architecture, rocks, water, ruins, mammal, stone, structure, mountains (`MargCIE-3×3`)
architecture, building, vegetation, mammal, animal, nature, water, wildlife, *winter* (`Tamura-3×3`)
winter, rocks, river, water, park, animal, wildlife, vegetation, mountain (`Log-Gabor`)
sun
sunset, dusk, twilight, dawn, silhouette, evening, nightfall, landscape, clouds
sunset, dusk, twilight, nightfall, dawn, silhouette, detail, closeup, texture
sunset, dusk, dawn, beach, clouds, sky, horizon, twilight, evening
sunset, dusk, silhouette, dawn, cloudy, evening, twilight, sunrise, mist
sunset
dusk, *sun*, dawn, *twilight*, silhouette, *nightfall*, clouds, *evening*, landscape
dusk, *sun*, *twilight*, *nightfall*, dawn, detail, silhouette, city, reflection
dusk, *sun*, dawn, horizon, beach, sky, clouds, *twilight*, silhouette
dusk, *sun*, dawn, silhouette, cloudy, *evening*, sunrise, *twilight*, mist
tree
vegetation, *nature*, wildlife, architecture, building, *forest*, animal, stone, rocks
vegetation, *nature*, wildlife, animal, water, *forest*, building, detail, birds
leaves, *vegetation*, nature, detail, stone, rock, *plants*, animal, wildlife
leaves, *forest*, *vegetation*, *plants*, nature, architecture, wildlife, park, stone
underwater
fish, nature, closeup, vegetation, plant, leaves, wildlife, rock, animal
fish, closeup, texture, plant, leaves, nature, detail, agriculture, vegetation
fish, leaves, nature, closeup, flora, detail, animal, rock, wildlife
fish, nature, mammal, animal, wildlife, rock, birds, vegetation, plant
vegetable
food, fruit, plant, closeup, flower, leaves, nature, detail, flora
food, fruit, plant, leaves, insect, flower, plants, meal, table
food, fruit, detail, pebbles, texture, table, leaves, closeup, agriculture
fruit, flower, closeup, *food*, plant, animal, face, birds, bird
vegetation
nature, *grass*, wildlife, park, animal, *tree*, water, rocks, rock
nature, *tree*, detail, *forest*, water, park, wildlife, *grass*, animal
nature, *grass*, rock, *leaves*, *tree*, detail, rocks, animal, *plants*
nature, *grass*, wildlife, park, rocks, rock, stone, mammal, animal
window
architecture, *building*, room, entrance, *house*, wall, tree, street, stone
detail, ornate, rock, wall, stone, room, *architecture*, tree, wood
detail, room, *architecture*, *building*, street, wall, nature, food, tree
architecture, *building*, entrance, room, temple, *house*, church, door, street
winter
snow, mammal, animal, wildlife, stone, architecture, building, rocks, water
snow, mammal, sand, stone, rocks, water, architecture, structure, animal
snow, wildlife, mammal, nature, vegetation, animal, leaves, rocks, tree
snow, park, wildlife, vegetation, animal, stone, rocks, mammal, castle
woods
forest, *tree*, floor, *vegetation*, nature, park, autumn, stone, wall
forest, *tree*, *vegetation*, floor, nature, wildlife, animal, park, grass
forest, autumn, garden, floor, *plants*, moss, leaves, *vegetation*, wall
forest, floor, autumn, *plants*, *tree*, leaves, garden, wall, architecture

Table A.6: Within-dataset keyword distribution similarities for Corel-16k, part III

Getty keyword w	Corel keyword w'
Cityscape	city
Clear Sky	sky
Cloud	clouds
Dusk	dusk
Field	field
Flower	flower
Fog	fog
Food	food
Grass	grass
Horizon	horizon
Landscape	landscape
Leaf	leaf
Lush Foliage	vegetation
Mammal	mammal
Mountain	mountain
Night	night
Non-Urban Scene	nature
One Animal	animal
One Person	person
other	other
Plant	plant
River	river
Sea	sea
Sky	sky
Skyline	skyline
Skyscraper	building
Snow	snow
Sun	sun
Sunset	sunset
Tree	tree
Underwater	underwater
Urban Scene	city
Vegetable	vegetable
Window	window
Winter	winter
Woods	woods

Table A.7: Keyword associations between modified Getty and Corel-16k

Web image keyword w	Corel keyword w'
Aerial view	aerial
Building exterior	building
Flower	flower
Jug	drink
Mugshot	face
Clouds	clouds
Crowd	people
Grazing animal	animal
Mountain	mountain
Sunset	sunset
other	other
Underwater fish	underwater

Table A.8: Keyword associations between Web images and Corel-16k

Keyword	Substituted density A.P.	Original density A.P.	Random
Aerial view	0.0413	0.3433	0.0099
Building exterior	0.0592	0.2553	0.0105
Flower	0.0225	0.2023	0.0066
Jug	0.0189	0.3636	0.0088
Mugshot	0.0563	0.4449	0.0106
Clouds	0.1327	0.3894	0.0046
Crowd	0.0066	0.1874	0.0051
Grazing animal	0.0081	0.2632	0.0039
Mountain	0.0804	0.3735	0.0060
Sunset	0.1334	0.6150	0.0065
Underwater fish	0.0277	0.2373	0.0091
Average	0.0534	0.3341	0.0074

Table A.9: Effects of substituting Corel-16k keyword models for annotating the Web images dataset

Keyword	Substituted density A.P.	Original density A.P.	Random
Building Exterior	0.1763	0.2130	0.0740
Cityscape	0.0874	0.1531	0.0201
Clear Sky	0.1413	0.3122	0.0466
Cloud	0.2468	0.2916	0.0762
Dusk	0.1194	0.1913	0.0443
Field	0.1820	0.2533	0.0316
Flower	0.0812	0.1658	0.0317
Fog	0.0965	0.1436	0.0124
Food	0.0961	0.4358	0.0389
Grass	0.1693	0.2215	0.0364
Horizon	0.1059	0.1670	0.0389
Landscape	0.0661	0.1448	0.0296
Leaf	0.1134	0.2110	0.0243
Lush Foliage	0.0896	0.2596	0.0121
Mammal	0.0473	0.3227	0.0229
Mountain	0.1028	0.1970	0.0347
Night	0.2967	0.3406	0.0661
Non-Urban Scene	0.0895	0.1741	0.0597
One Animal	0.2077	0.3482	0.1088
One Person	0.1843	0.3289	0.0579
Plant	0.0698	0.1764	0.0450
River	0.0341	0.1753	0.0227
Sea	0.0690	0.1861	0.0465
Sky	0.2091	0.2804	0.0839
Skyline	0.0829	0.1709	0.0163
Skyscraper	0.0671	0.1391	0.0217
Snow	0.0989	0.1861	0.0348
Sun	0.0923	0.1510	0.0155
Sunset	0.1454	0.1988	0.0209
Tree	0.1219	0.2165	0.0718
Underwater	0.1613	0.2451	0.0230
Urban Scene	0.1607	0.1912	0.0670
Vegetable	0.0289	0.1777	0.0159
Window	0.0907	0.1772	0.0370
Winter	0.0764	0.1628	0.0271
Woods	0.1112	0.1686	0.0146
Average	0.1200	0.2188	0.0397

Table A.10: Effects of substituting Corel-16k keyword models for annotating the modified Getty collection