Content-Based Hierarchical Classification of Vacation Images

Aditya Vailaya, [+]Mário Figueiredo,[*] Anil Jain, [#]HongJiang Zhang

Dept. of Comp. Sc. & Eng.	[+] Instituto de Telecomunicações
Michigan State University	Instituto Superior Técnico
East Lansing, MI 48824, USA	1096 Lisboa Codex, Portugal

[#] Internet Systems & Applications Lab
Hewlett Packard Labs
Palo Alto, CA 94304, USA

vailayaa@cps.msu.edu, [+]mtf@lx.it.pt, jain@cps.msu.edu, [*]hjzhang@hpl.hp.com

Key words: Content-based indexing/retrieval, Image content analysis, Digital libraries.

Abstract

Grouping images into (semantically) meaningful categories using low-level visual features is a challenging and important problem in content-based image retrieval. Using binary Bayesian classifiers, we show how high-level concepts can be understood from low-level images under the constraint that the image does belong to one of the classes in question. Specifically, we consider the hierarchical classification of vacation images; at the highest level, images are classified into indoor-outdoor classes, outdoor images are further classified into city-landscape classes, and finally, a subset of landscape images is classified into sunset, forest, and mountain classes. We demonstrate that a small codebook (the optimal size of codebook is selected using MDL principle) extracted from a vector quantizer can be used to estimate the class-conditional densities of the observed features needed for the Bayesian methodology. The classifiers have been built on a database of $6,931$ vacation photographs. Our system achieved an accuracy of 90.8% for indoor-outdoor classification, 94.3% for city vs. landscape classification, 94.9% for sunset vs. forest & mountain classification, and 93.6% for forest vs. mountain classification. Our final goal is to combine multiple 2-class classifiers into a single hierarchical classifier.

[*] M. Figueiredo was partially supported by Nato Grant *NATOCRG 960010* & Portuguese PRAXIS XXI program, grant no. 2/2.1/T. I. T./1580/95.

1 Introduction

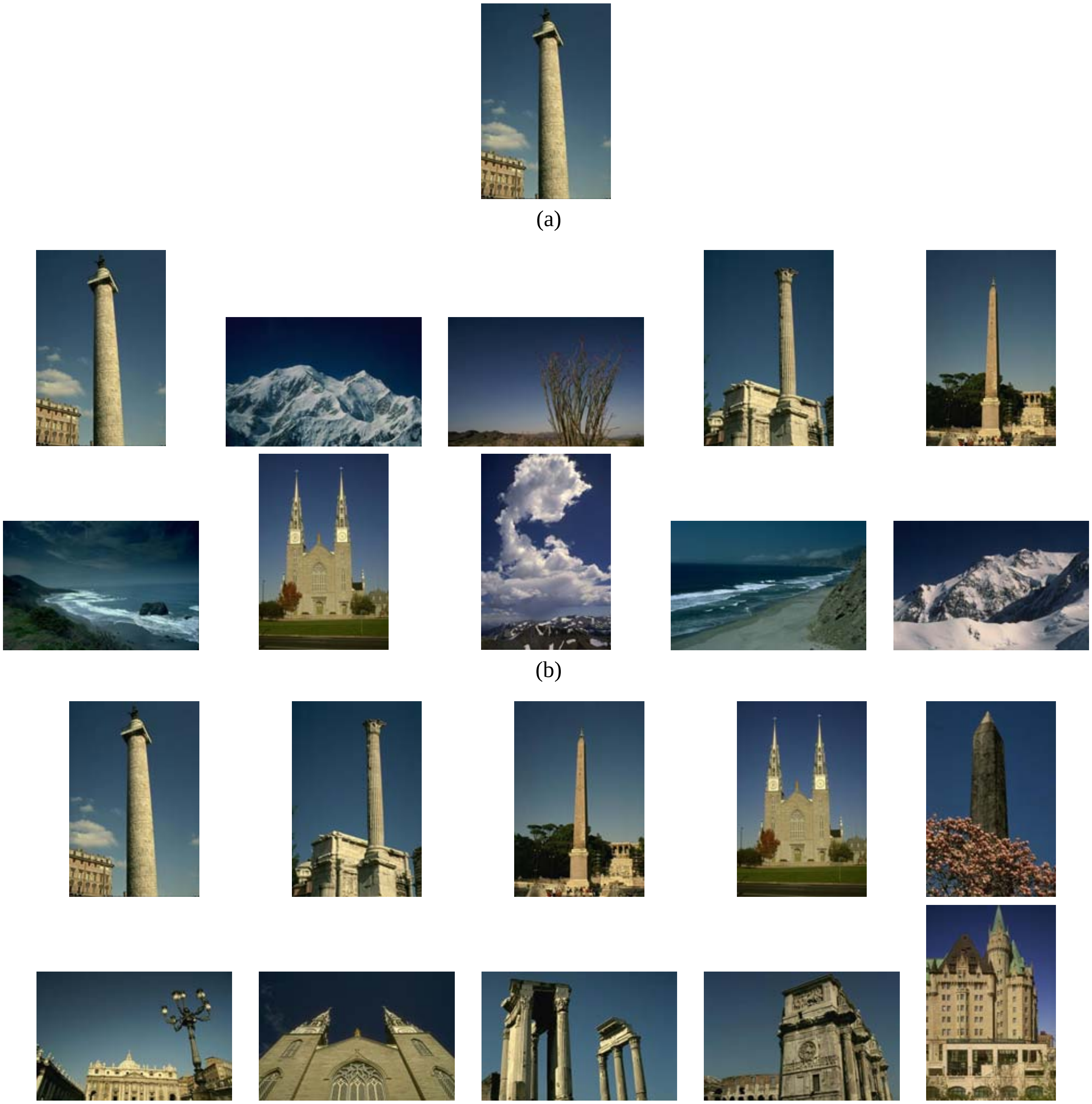
Content-based image organization and retrieval has emerged as an important area in computer vision and multimedia computing. This is mainly driven by technological breakthroughs which allow us to digitize, store, and transmit images in a very cost effective and efficient manner. A large number of commercial organizations have large image and video collections of programs, news segments, games, paintings, and artifacts that are being digitized. Organizing these image and video libraries into a small number of categories and providing effective indexing is imperative for accessing, browsing, and retrieving useful data in "real-time". With the development of digital photography, more and more people are able to store their vacation and personal photographs on their computers. Travel agencies are interested in digital archives of photographs of holiday resorts. A user could query these databases to plan a vacation. These digital databases are not a dream of the future, but have become a reality. However, in order to make these databases more useful, we need to develop schemes for indexing and categorizing the humungous data.

Due to the limitation of textual features as indices to image databases, there has been an intense activity in developing image retrieval methods based on image content. Various systems have been proposed for content-based image retrieval, such as QBIC [1], Photobook [2], FourEyes [3], SWIM [4], Virage [5], Visualseek [6], Netra [7], and MARS [8]. These systems follow the paradigm of representing images using a set of image attributes, such as color, texture, shape, and layout which are archived along with the images in the database. A retrieval is performed by matching the feature attributes of a query image with those of the database images. Users typically do not think in terms of low-level image features while querying digital databases, i.e., user queries are typically based on semantics (e.g., show me a sunset image) and not on low-level image features (e.g., show me a predominantly red and orange image). As a result, most of these image retrieval systems have poor performance for specific queries. For example, Figure 1(b) shows the top-10 retrieved results (based on color histogram features) for the query in Figure 1(a) on a database of $2,145$ images of city and landscape scenes. While the query image has a specific monument (a tower here), some of the retrieved images include scenes of mountains and coasts. A successful grouping of these database images into semantically meaningful classes can greatly enhance the performance of a content-based image retrieval system. Figure 1(c) shows the top-10 retrieved results (again based on color histogram features) on a database of 760 *city* images for the same query; clearly, filtering out landscape images from the image database prior to querying improves the retrieval result.

As we have shown in Figs. 1 (a)-(c), a successful indexing and categorization of images will greatly enhance the performance of content-based image retrieval systems by filtering out images from irrelevant classes during matching. This rather difficult problem has not been adequately addressed in current image database systems. The main challenge is to group images into semantically meaningful categories (or index images in a database) based on low-level visual features of the images. One attempt to solve this problem is the hierarchical indexing scheme proposed by Zhang and Zhong [9, 10], which uses a Self-Organization Map (SOM) to perform clustering based on color and texture features. This indexing scheme was further applied in [11] to create a texture thesaurus for indexing a database of large aerial photographs. However, the success of such clustering-based indexing schemes is often limited, largely due to the low-level feature-based representation of image content. For example, Figures 2 (a)-(d) show two images (a fingerprint and a landscape image) and their corresponding edge direction coherence feature vectors (these features are defined in [12]). Although, the two images denote two very different concepts, their low-level edge features are highly similar; the Euclidean distance between the corresponding histograms is only 0.0147 (distances in range [0,1]). This shows the limitations of low-level features in capturing semantic content in an image. Yet, as we shall show later, in constrained environments, these very low-level features can be used to discriminate between conceptual image classes. To achieve the goal of automatic categorization and indexing of images in a large database, we need to develop robust schemes to identify salient image features that capture a certain aspect of semantic content of these images. In other words, we first need to specify/define pattern classes, so that the database images can be organized in a *supervised* fashion.

In this paper, we pose the image classification problem in a Bayesian framework. Specifically, we address the problem of classifying vacation photographs. Our early experiments with 8 human subjects [12] on a database of 171 vacation images revealed the classification hierarchy as shown in Figure 3(a). A total of 11 semantic categories were revealed from these experiments. The first four classes of forests, mountains, beach scenes, and pathways were grouped into the class, *natural scenes*. The cluster of natural scenes and the sunset images were further grouped into the *landscape* class. The clusters of city shots, monuments, and shots of Washington DC were grouped into *city* class. Finally, the miscellaneous, face, landscape, and city classes were grouped into the top-level class of *vacation* scenes. It is these groupings, which motivated us to address the issue of hierarchical classification of vacation images.

Our goal is to develop a hierarchical classifier similar to the one shown in Figure 3(a) for vacation images. To

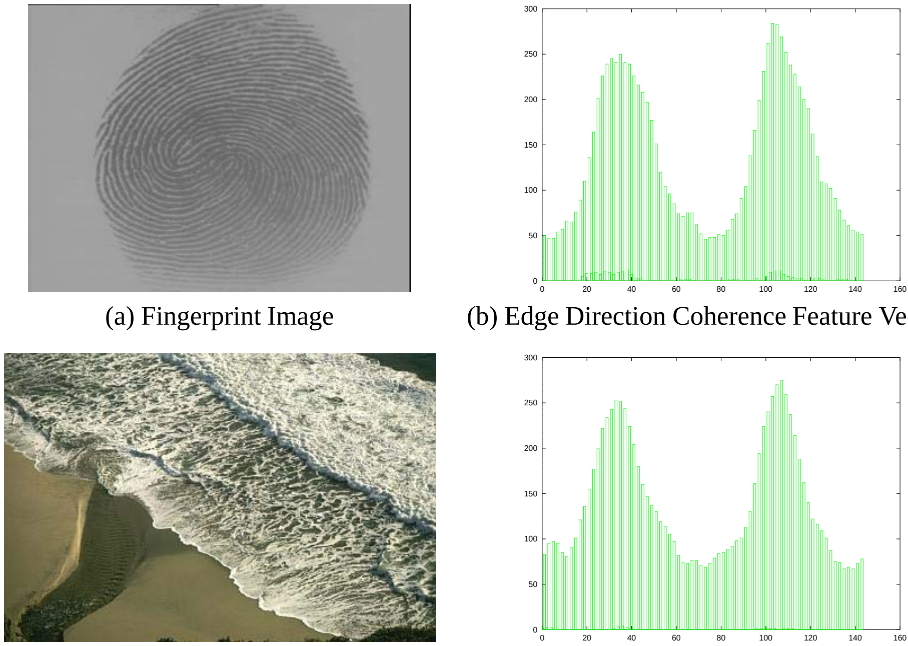


(a)

(b)

(c)

Figure 1: Color-based retrieval results; (a) query image; (b) top-10 retrieved images from 2, 145 city and landscape images; (c) top-10 retrieved images from 760 city images; clearly filtering out landscape images prior to querying improves the retrieval results.



(a) Fingerprint Image (b) Edge Direction Coherence Feature Vector

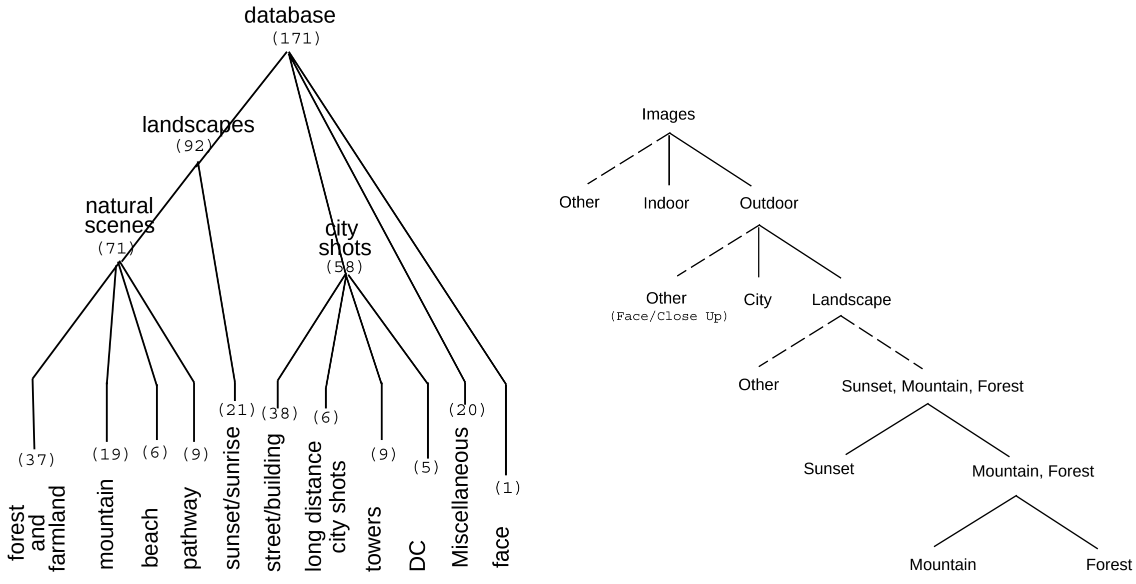
(c) Landscape Image (d) Edge Direction Coherence Feature Vector

Figure 2: Edge direction coherence vector features for (a) fingerprint and (c) landscape image; The difference, $d(b, d)$, between these two histograms is 0.0147.

make the problem more tractable, we simplified the classification hierarchy as shown in Figure 3(b). The solid lines show the classification problems addressed in this paper. At the highest level, vacation images can be divided into *indoor*, *outdoor*, and *other* scenes. Outdoor shots can then be further classified into *city*, *landscape*, and *other* classes. Landscape images can then be dichotomized into *sunset*, *mountain*, *forest*, and *other* classes. While the above hierarchy is not in itself complete (a user may be interested in querying the database for images captured in the evening - (day/night classification), images containing faces (face vs. non-face classification), or images containing text (text vs. non-text classification)), it is a reasonable approach to simplify the image retrieval problem.

The four classification problems that we have addressed in the hierarchy are: (i) indoor vs. outdoor classification; (ii) city vs. landscape classification; (iii) sunset vs. forest & mountains classification; and (iv) forest vs. mountains classification. The indoor vs. outdoor classification problem can be stated as follows: Given an image, classify it as either an indoor or an outdoor image. Most of the images can be classified into either of the two classes. Exceptions include close-up shots, pictures of a window or door, etc. Outdoor images can be further classified into city vs. landscape images [12, 13]. City vs. landscape classification problem can be posed as follows: Given an outdoor image, classify it as either a city or a landscape image. City scenes can be characterized by the presence of man-made objects and structures such as buildings, cars, roads, etc. Natural scenes, on the other hand, lack these structures. A subset of landscape images can be further classified into one of the sunset, forest, and mountain classes. Sunset scenes can be characterized by saturated colors (red, orange, or yellow), forest scenes have predominantly green color distribution due to the presence of dense trees and foliage, and mountain scenes can be characterized by long distance shots of mountains (either snow covered, or barren plateaus). We assume that the input image does belong to one of the classes under consideration. This restriction is posed because, automatically rejecting images not belonging to the specific classes (such as city or landscape) based on low-level image features alone is in itself a very difficult problem (see Figure 2).

The classification problems formulated above can be addressed using Bayes decision theory. The probabilistic models required for the Bayesian approach are estimated during a training phase; in particular, the class-conditional probability density functions of the observed features are estimated under a Vector Quantization (VQ) framework [14, 15, 16]. The Bayesian approach has the following advantages: (i) it not only provides a classification rule, but also assigns a degree of confidence in the classification; (ii) a small number of codebook vectors represent a particular class of images, thereby greatly reducing the number of comparisons necessary for each classification; and (iii) it naturally



(a) (b)

Figure 3: Experiments with human subjects: (a) A hierarchical organization of the 11 categories obtained from groupings provided by human subjects [12]; (b) Simplified semantic classification of images; solid lines show the classification problems addressed in this paper.

allows for the integration of multiple features through the class-conditional densities.

The paper is organized as follows. In Section 2 we discuss the Bayesian framework for image classification. An introduction to Vector Quantization (VQ) and density estimation is presented in Section 3. The experimental results are described in Section 4. Section 5 finally concludes the paper and presents directions for future research.

2 Bayesian Framework

Bayesian theory provides a formal (probabilistic) framework for image classification problems. It requires that all assumptions be explicitly specified to build models (observation model, prior, loss function) which are then used to derive an “optimal” decision/classification rule. Optimality here means that, under the assumed models, there does not exist any other classification rule which has a lower expected loss. The Bayesian paradigm has been successfully adopted in a number of image analysis and computer vision (both low and high level) problems, such as restoration, segmentation, and classification (see [17, 18] and the references therein). However, its use in content-based retrieval from image databases is just being realized [15].

2.1 Basic Elements

The Bayesian framework requires that all the entities involved in decision making be adequately formalized:

- Each observed image $\boldsymbol{x}$ belongs to a set $\mathcal{I}$ of possible images.
- The set $\mathcal{I}$ is assumed to be partitioned into K classes $\Omega = \{\omega_1, \omega_2, ...\omega_K\}$; these classes are exhaustive and mutually exclusive, i.e., any image $\boldsymbol{x}$ from $\mathcal{I}$ belongs to one and only one class.
- Each observed image $\boldsymbol{x}$ is modeled as a sample of a random variable $\mathbf{X}$, whose class-conditional probability density function for class ω_n is written as $f_{\mathbf{X}}(\boldsymbol{x}|\omega_n)$.
- An *a priori* knowledge concerning the classes is expressed via a probability function defined on the set of classes, $\{p(\omega_1), p(\omega_2), ..., p(\omega_K)\}$.

- A loss function, $\mathcal{L}(\omega, \hat{\omega}) : \Omega \times \Omega \to \mathcal{R}$, specifying the loss incurred when class $\hat{\omega}$ is chosen and the true class is ω. As is common in classification problems, we adopt the "$0/1$" loss function; $\mathcal{L}(\omega, \omega) = 0$, and $\mathcal{L}(\omega, \hat{\omega}) = 1$, if $\omega \neq \hat{\omega}$.
- Finally, the solution of the classification problem is a decision rule $\delta(\boldsymbol{x}) : \mathcal{I} \to \Omega$ which maps any possible observed image into one of the available classes.

2.2 Image Features

In many image analysis problems, it is typical that the classification is based on, say, m features extracted from the observed image, rather than directly on the raw pixel values. Let $\boldsymbol{y} = \{\boldsymbol{y^{(1)}}, \boldsymbol{y^{(2)}}, \ldots, \boldsymbol{y^{(m)}}\}$ denote the set of m features based on which the classification procedure must operate. For computational simplicity it is typical to assume that the features are conditionally independent. As a result, the class-conditional density functions can be written as

$$f_{\mathbf{X}}(\boldsymbol{x} \mid \omega) \equiv f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega) = \prod_{i=1}^{M} f_{\mathbf{Y}^{(\mathrm{i})}}(\boldsymbol{y^{(i)}} \mid \omega). \tag{1}$$

The classification problem can be stated as: "given a set of observed features, $\boldsymbol{y}$, from an image $\boldsymbol{x}$, classify $\boldsymbol{x}$ into one of the classes in Ω.

2.3 Classification Rule

In the Bayesian framework, all inferences are based on the *a posteriori* probability function, which is obtained by combining the class-conditional observation models with the *a priori* class probabilities. This is done via Bayes law

$$p(\omega \mid \boldsymbol{y}) = \frac{f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega)\, p(\omega)}{f_{\mathbf{Y}}(\boldsymbol{y})}, \tag{2}$$

where the denominator, $f_{\mathbf{Y}}(\boldsymbol{y})$, in Eq. (2) is the unconditional (or marginal) probability density function of the observed features, which serves simply as a normalizing constant.

The adopted "$0/1$" loss function leads to what is the most common criterion in Bayesian classification problems: choose the class whose *a posteriori* probability is maximum. This is known as the *maximum a posteriori* (MAP) criterion, and is given by

$$\hat{\omega} = \delta(\boldsymbol{x}) = \arg\max_{\omega \in \Omega} \{p(\omega \mid \boldsymbol{y})\} = \arg\max_{\omega \in \Omega} \{f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega)\, p(\omega)\}. \tag{3}$$

In addition to reporting the MAP classification of a given image, say ω_k, the Bayesian approach also assigns a degree of confidence to that classification, which is proportional to $p(\omega_k \mid \boldsymbol{y})$. We next describe a procedure to estimate the class-conditional density functions.

3 Density Estimation Using Vector Quantization

The performance of the Bayes classifier clearly depends on the ability of the feature set $\boldsymbol{y}$ to discriminate among the various classes. Moreover, since the class-conditional densities have to be estimated from training data, the accuracy of these estimates is also critical. Choosing the right set of features for a given classification problem is a difficult problem and we do not discuss the issue here. We concentrate instead on estimating the class-conditional densities for which we adopt a Vector Quantization (VQ) based approach [16].

3.1 Introduction to Vector Quantization

Vector Quantization (as its name implies) is a compression/quantization technique that is applied to vectors rather than scalars. Just like scalar measurements can be quantized by rounding off or setting thresholds, VQ quantizes a group of numbers (components of a vector) together. Thus, VQ takes as input a p-dimensional vector and quantizes it into a p-dimensional *reproduction vector*. A VQ can be specified by a set of reproduction vectors and a rule for mapping input vectors to the reproduction vectors.

In the compression and communication applications, a Vector Quantizer is described as a combination of an encoder and a decoder. A p-dimensional VQ consists of two mappings: an encoder γ which maps the input alphabet ($\mathbf{A}$) to the channel symbol set ($\mathbf{M}$), and a decoder β which maps the channel symbol set ($\mathbf{M}$) to the output alphabet ($\hat{\mathbf{A}}$), i.e., $\gamma(\boldsymbol{y}) : \mathbf{A} \to \mathbf{M}$ and $\beta(\boldsymbol{v}) : \mathbf{M} \to \hat{\mathbf{A}}$. A distortion measure $\mathcal{D}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ specifies the cost associated with quantization, where $\hat{\boldsymbol{y}} = \beta(\gamma(\boldsymbol{y}))$. Usually, an optimal quantizer minimizes the average distortion under a size constraint on $\mathbf{M}$. The generalized Lloyd algorithm for vector quantization uses the mean square error (MSE) criterion for distortion and is equivalent to a K-means clustering algorithm [19], where K is the size of the output alphabet, $\hat{\mathbf{A}} : \{\hat{\boldsymbol{y}}_{\boldsymbol{i}}, i = 1, \ldots, K\}$. An input vector $\boldsymbol{y} \in \mathbf{A}$ is quantized into one of the K output vectors $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$, also referred to as *codebook vectors*, such that

$$\mathcal{D}(\boldsymbol{y}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}) \leq \mathcal{D}(\boldsymbol{y}, \hat{\boldsymbol{y}}_{\boldsymbol{j}}), \ \forall\, 1 \leq j \leq K. \tag{4}$$

These codebook vectors define a partition of the feature space, according to Eq. (4), into the so-called Voronoi cells, $\{S_i, i = 1, 2, \ldots, K\}$. Figure 4 shows an example of such a 2-D Voronoi tessellation where the $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ are shown as square dots. As the data points get closer, the cells become more compact. According to Eq. (4), an input vector is assigned the codebook vector of the cell it falls into. A comprehensive study of VQ, choice of distortion measures, and use of VQ in classification and compression is presented in [20, 16].

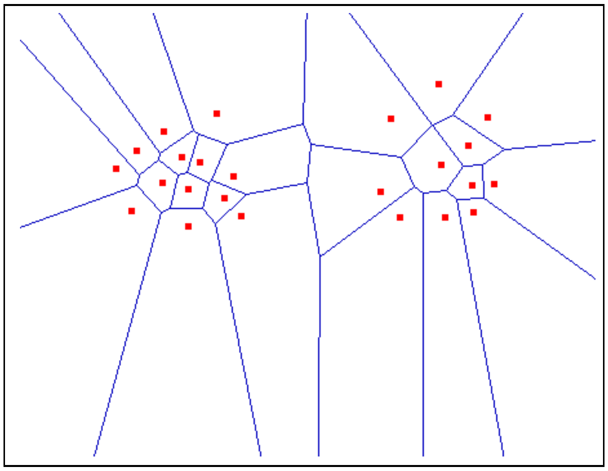


Figure 4: Voronoi Tessellation for 2-D data points in two clusters.

3.2 VQ as a Density Estimator

Vector quantization provides an efficient tool for density estimation [16]. Consider n training samples from a class ω. In order to estimate the class-conditional density of the feature vector $\boldsymbol{y}^{(\boldsymbol{i})}$ given the class ω, i.e., $f_{\mathbf{Y}^{(\mathrm{i})}}(\boldsymbol{y}^{(\boldsymbol{i})} \mid \omega)$, a vector quantizer is used to extract q (with $q < n$, hopefully $q \ll n$) codebook vectors, $\mathbf{v}_{\mathbf{j}}^{(\mathbf{i})}$ $(1 \leq j \leq q)$, from the n training samples. It has been shown (see [16]) that in the so-called high-resolution approximation (i.e., for sufficiently small Voronoi cells), the class-conditional density can be approximated as a piecewise-constant function over each cell $S_j^{(i)}$, with value

$$f_{\mathbf{Y}^{(\mathrm{i})}}(\boldsymbol{y}^{(\boldsymbol{i})} \mid \omega) \approx \frac{m_j^{(i)}}{Vol(S_j^{(i)})}, \tag{5}$$

where $m_j^{(i)}$ and $Vol(S_j^{(i)})$ are the ratio of training samples falling into cell $S_j^{(i)}$ and the volume of the cell $S_j^{(i)}$, respectively. This approximation fails if the Voronoi cells are not sufficiently small, as is the case when the dimensionality of the feature vector $\boldsymbol{y}^{(\boldsymbol{i})}$ is large. The class-conditional densities can then be approximated using a kernel-based approach [16, 15], approximating the density by a mixture of Gaussians, each centered at a codebook vector. In most VQ algorithms, the codebook vectors are iteratively selected by minimizing the MSE (mean square error) which is the sum of the Euclidean distances of each training sample from its closest codebook vector. Hence, an identity covariance matrix can be assumed for the Gaussian components used to represent the densities [15], resulting in the following (approximate) class-conditional densities:

$$f_{\mathbf{Y}^{(\mathrm{i})}}(\boldsymbol{y^{(i)}} \mid \omega) \propto \sum_{j=1}^{q} m_j^{(i)} * \exp(-\|\boldsymbol{y^{(i)}} - \boldsymbol{v_j^{(i)}}\|^2/2). \tag{6}$$

A more comprehensive approach would be to use the Mahalanobis distance [19] in estimating the codebook vectors; but, if feature dimensionality is high and the number of training samples is small, the estimated covariance matrices are likely to be singular.

3.3 Selecting Codebook Size

A key issue in using vector quantization for density representation is the choice of the codebook size. It is clear that, given a training set, the VQ-approximated likelihood (probability) of that training set will keep increasing as the dimension of the codebook grows; in the limit, we would have a code vector for each training sample, with the corresponding probability equal to one. To address this issue, we adopt the *minimum description length* (MDL) principle [21]. MDL is an information-theoretic criterion which has recently been used for several problems in computer vision and image processing (see [22], and references therein). We start by noting that the VQ learning algorithm basically looks for the *maximum likelihood* estimates of the parameters of the mixture in Eq. (6). The first key observation behind MDL is that looking for an ML estimate is equivalent to looking for the Shannon code for which the observations have the shortest code-length [21]; this is so because Shannon's optimal code-length[1], for some set of observations, $\mathcal{Y} : \{\boldsymbol{y^{(i)}}(1), \ldots, \boldsymbol{y^{(i)}}(n)\}$, obeying some joint probability density function $f(\mathcal{Y}|\boldsymbol{\theta}_{(q)})$, is simply [23, 22]

$$L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) = -\log f(\mathcal{Y}|\omega, \boldsymbol{\theta}_{(q)}). \tag{7}$$

Under the assumption of independent samples, $\boldsymbol{y^{(i)}}(j)$ $(1 \leq j \leq n)$, the joint likelihood in Eq. (7) can be written as

$$L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) = -\sum_{j=1}^{n} \log f(\boldsymbol{y^{(i)}}(j)|\omega, \boldsymbol{\theta}_{(q)}); \tag{8}$$

in our case, each of the marginals in the above likelihood is the one in Eq. (6) and $\boldsymbol{\theta}_{(q)}$ contains the codebook vectors, $\{\mathbf{v_j^{(i)}} \ : \ 1 \leq j \leq q\}$, and the weights $m_j^{(i)}$. The second fundamental fact is that the parameters themselves are also part of the code, in the following sense: a code word representing $\mathcal{Y}$ can not be decoded by itself; only a full knowledge of $f(\mathcal{Y}|\boldsymbol{\theta}_{(q)})$ (i.e., of its parameters) allows reconstructing the code and respective decoder. Accordingly, the MDL criterion states that the description code-length to be minimized by the estimate must include not only the data code-length but also the code-lengths of the parameters. The resulting criterion for the choice of q (codebook size) is then

$$\widehat{q} = \arg\min_{q} \left\{ L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) + L(\boldsymbol{\theta}_{(q)}) \right\}. \tag{9}$$

Finally, concerning the parameters description length, $L(\boldsymbol{\theta}_{(q)})$, the commonly adopted choice is $L(\boldsymbol{\theta}_{(q)}) = (\zeta(q)/2)\log n$, where n is the sample size and $\zeta(q) = q + q \dim(\boldsymbol{y^{(i)}})$ is the number of real-valued parameters needed to specify a q^{th}-order model and $\dim()$ represents the dimension of the feature space [21]. This is an asymptotically optimal choice, which is only valid when all the parameters depend on all the data, which is not the case in the present problem. The weights $m_j^{(i)}$ are, in fact, estimated from all the data; however, each $\mathbf{v_j^{(i)}}$ is estimated from the $m_j^{(i)}$ samples that fall in the associated cell. Accordingly, we use the following parameter description length

$$L(\boldsymbol{\theta}_{(q)}) = \frac{q}{2}\log n + \frac{\dim(\boldsymbol{y^{(i)}})}{2} \sum_{j=1}^{q} \log(m_j^{(i)}\, n), \tag{10}$$

where the first term accounts for the weights m_j while the second one corresponds to the codebook vectors themselves.

[1] In bits or *nats*, if base-2 or natural logarithms are used, respectively [23].

4 Experimental Results

The Bayesian paradigm was applied to generate a hierarchical classification of vacation photographs. Images were first classified into indoor and outdoor classes. Outdoor images were then classified into city and landscape classes. Finally, a subset of landscape images was classified into sunset, forest, and mountain classes. Experiments were conducted on two databases (both independently and combined) of $5,081$ (indoor vs. outdoor classification) and $2,716$ (city vs. landscape classification and further classification of landscape images) images. The two databases, henceforth referred to as database D1 and database D2, had 866 images in common, thus the entire database contains $6,931$ images. These images were collected from various sources (Corel stock photo library, scanned personal photographs, key frames from digitized video of television serials, and images downloaded from the web) and are of varying sizes (from 150×150 to 750×750). The color images are represented by 24-bits per pixel and stored as JPEG images. The ground truth for all the images was assigned by a single subject. We next describe the features that were extracted from the images.

4.1 Image Features

The accuracy of the Bayesian classifiers depends on the underlying low-level representation of the images. The more discriminative the features, better is the classification accuracy and using just any feature will not yield good classification results. For example, outdoor images tend to have uniformity in spatial color distributions, such as the sky is on top and typically blue in color. Indoor images tend to have more varied color distributions and have more uniform lighting (most are close up shots). Thus, it seems logical to use spatial color distribution as a feature for discriminating between indoor and outdoor images. On the other hand, shape features may not be useful, since similar shapes and objects (people, furniture, plants, edges due to walls, etc.) can be present in both indoor and outdoor images. We thus, use spatial color information features that represent these qualitative attributes of indoor and outdoor classes. Specifically, first and second order color moments in the LUV color space were used as color features (it was pointed out in [24] that color moments in the LUV color space yielded better results during image retrieval than color moments in other color spaces). The image was divided into 10×10 sub-blocks and six features (3 each for mean and standard deviation) were extracted. As another set of features for indoor vs. outdoor classification, we extract sub-block MRSAR features as described in [25].

We look for similar qualitative attributes in city vs. landscape classification problem, and further classification of landscape images. City images usually have strong vertical and horizontal edges due to the presence of man-made objects. Non-city (natural) images tend to have edges randomly distributed in all directions. A feature based on the distribution of edge directions can discriminate between these two categories of images [12]. On the other hand, color features would not have sufficient discriminatory power as man-made objects have arbitrary color distributions (two buildings need not have the same color). In the case of classification of landscape images into further categories, such as sunset, forest, and mountain, global color distributions seem to adequately describe these classes. Sunset pictures typically have highly saturated colors (mostly yellow and red); mountain images tend to have a sky in the background (typically blue); and forest scenes tend to have more greenish distributions (presence of dense foliage). Based on the above observations, we use edge direction features (histograms and coherence vectors) for city vs. landscape classification and color features (histograms and coherence vectors) in the HSV color space for further classification of landscape images [12]. Table 1 briefly describes the qualitative attributes of the various classes and the features used to represent them.

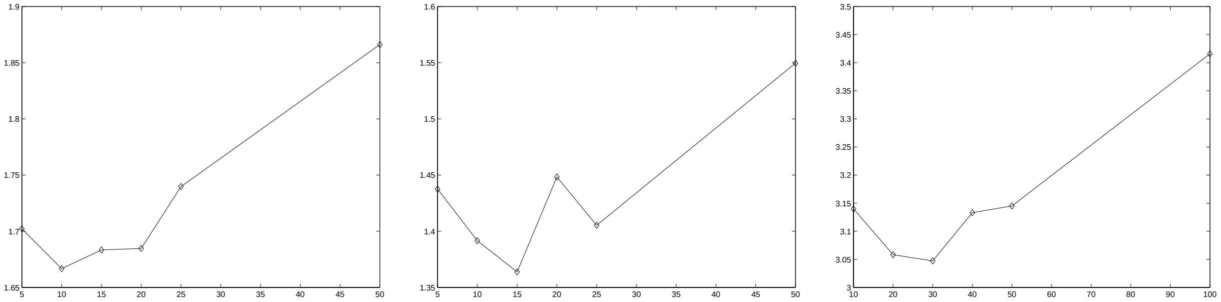
4.2 Vector Quantization

A number of experiments were conducted to study the robustness and limitations of the various classifiers. The LVQ_PAK package [26] was used for vector quantization. For every class, half of the database images were used to train the VQ for each of the image features. The MDL principle [21] described in Section 3.3 was used to determine the codebook size from the training samples for the various classifiers. We present the results of applying the MDL principle to the indoor vs. outdoor classifier for the spatial color moment features. Figures 5(a)-(c) show the plots of $L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) + L(\boldsymbol{\theta}_{(q)})$ (criterion to be minimized in Eq. (9)) vs. the codebook size, q, for the spatial color moment features for (a) indoor, (b) outdoor, and (c) both the classes. As can be seen in the figures, $q \sim 10$ minimizes the criterion in Eq. (9) for the indoor class, while $q \sim 15$ minimizes the criterion for the outdoor class. Combining the two yields $q \sim 30$ as the optimal number of codebooks for the indoor vs. outdoor classifier based on the training

Classification Problem	Qualitative Attributes	Low-level Features
Indoor vs. Outdoor	spatial color & lighting distributions	10×10 sub-block color moments in LUV space
City vs. Landscape	distribution of edges	edge direction histograms & coherence vectors
Sunset vs. Forest vs. Mountain	global color distributions & saturation values	color histograms & coherence vectors in HSV space

Table 1: Feature saliency: Qualitative attributes of various classification problems addressed in the paper and respective low-level features used for discrimination.

samples. Hence, 15 codebook vectors were extracted for both indoor and outdoor classes. Based on a similar analysis (see [13]), 20 codebook vectors were extracted for each of the city and landscape classes. For further classification of landscape images, a codebook of 5 vectors was selected for each class. The codebook vectors for each class were then stored as representatives for the class. Table 2 shows the number and dimensionality of the codebook vectors for the various classification problems using the features geared towards the classification (color-moments for indoor vs. outdoor, edge direction coherence vectors for city vs. landscape, and color coherence vectors for further classification of landscape images).



(a) (b) (c)

Figure 5: Determining codebook size for spatial color moment features for the indoor vs. outdoor classification problem; (a) indoor class; (b) outdoor class; (c) indoor and outdoor classes combined; x-axis represents the codebook size and y-axis represents the optimality criterion to be minimized.

Classification Problem	# of Codebook Vectors / Class	Feature Dimensionality
Indoor/Outdoor	15	600
City/Landscape	20	145
Sunset/Forest/Mountain	5	640

Table 2: Vector Quantization: # and dimensionality of codebook vectors extracted for the various classifiers.

4.3 Classification

Given an input image, the classifier compares the extracted features with the stored codebook vector features of a particular class (say, “city” class) and estimates the class-conditional probabilities for each of the features using Eq. (6). These probabilities are then used to estimate the a posteriori probability (Eq. (2)) that the image comes from the city class, given the extracted feature vectors. We present classification accuracies on a set of independent test patterns (hold-out error) as well as on the training patterns (re-substitution error). We have done the various classifications based on individual features and also based on a combination of various features. As we show later, each of the individual features chosen for the classification problems has sufficient discrimination power for that particular classification problem, and combining other features does not improve the results.

4.3.1 Indoor Vs. Outdoor Classification

Database D1 was used to train the indoor vs. outdoor classifier. This database consisted of $2,470$ indoor and $2,611$ outdoor images. Apart from the color moment features, we also used the sub-block MRSAR features [25], the edge direction features , and the color histogram features for the classification. MRSAR features yielded an accuracy of around 75% on the test set, edge direction and color histogram features yielded an accuracy of around 60%, and the color moment features yielded a much higher accuracy of around 90%. These results show that the spatial color distribution (probably capturing lighting effects) is more suited for indoor vs. outdoor classification. A combination of color and texture features did not yield a better accuracy than the color moment features. Table 3 shows the classification results for the color moment features for the indoor vs. outdoor classification problem. The classifier performed with an accuracy of 94.2% and 88.2% on the training set and an independent test set (Test1 in Table 3), respectively. On a different test set (Test2 in Table 3) of $1,850$ images from database D2, the classifier performed with an accuracy of 89.5%. The classifier performed with an overall accuracy of 90.8% on the entire database of $6,931$ images. Szummer et al. [25] and Yiu [27] report classification accuracies of approximately 90% on databases of size $1,324$ and 500 images, respectively. Thus, our classifier performance is comparable to those reported in the literature. A major advantage of the Bayesian classifier is its efficiency due to the small number of codebook vectors needed to represent the training data. A more detailed analysis of the Bayesian approach is presented in Section 4.4.

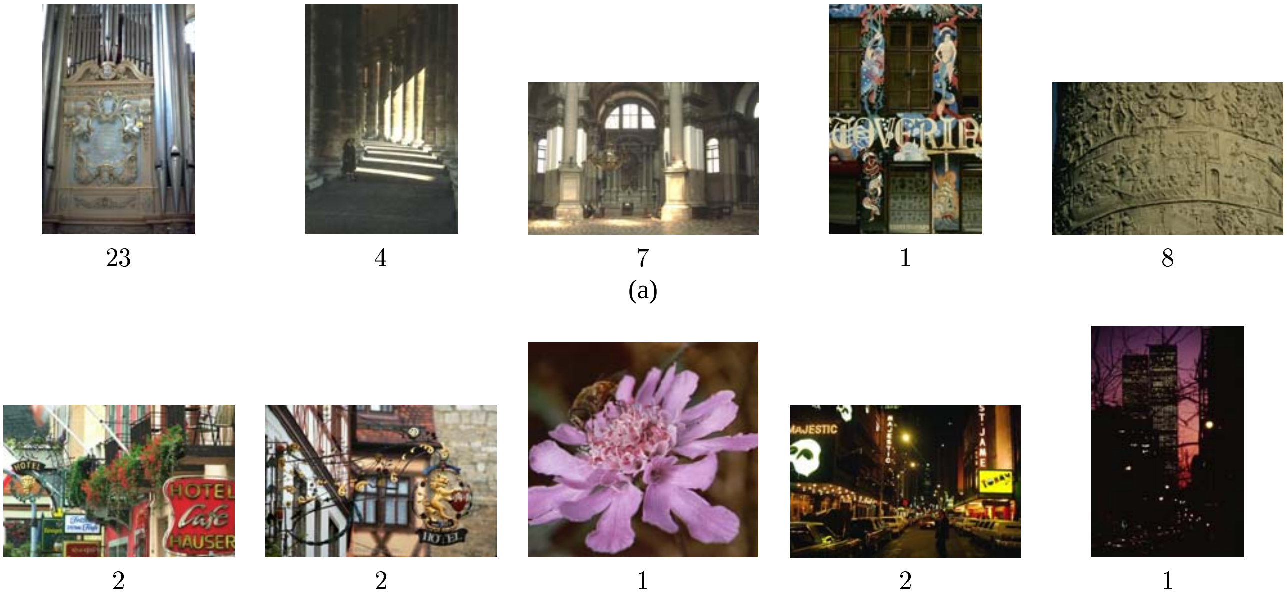
Figure 6 shows a representative subset of the misclassified indoor and outdoor scenes. Presence of bright shots either from some light source or from sunshine through windows and doors seems to be a main cause of misclassification of indoor images. The main reasons for the misclassification of outdoor images are as follows: (i) uniform lighting along the image mostly as a result of a close-up shot and (ii) dark images (some of the indoor images used in the training set were dark digital images and hence, most dark pictures are classified as indoor scenes). The results show that spatial color distribution captured in the sub-block color moment features has sufficient discrimination power for the indoor vs. outdoor classification problem.

Test Data	Data Size	Accuracy ($\%$)
Training Set	$2,541$	94.2
Test1	$2,540$	88.2
Test2	$1,850$	88.7
Entire Database	$6,931$	90.8

Table 3: Classification accuracies (in %) for color moment features for various test data; Test1 and Test2 are two independent test sets.

4.3.2 City Vs. Landscape Classification

The city vs. landscape classification problem and further classification of landscape images into sunset, forest, and mountain classes using the Bayesian framework has been addressed in detail in [13]. We present the gist of the results here. Table 4 shows the classification results for the city vs. landscape classification problem using database D2. Edge direction coherence vector provides the best individual accuracy of 95.2% for the training data and 92.4% for the test data. A total of 155 images were misclassified (a classification accuracy of 94.3%) when the edge direction coherence vector was combined with the color histogram feature. Figure 7 shows a representative subset of the misclassified city and landscape scenes. Most of the misclassifications for city images could be attributed to the following reasons: (i) long distance city shots at night (which made it difficult to extract edges), (ii) top view of city scenes (lack of vertical edges), (iii) highly textured buildings, and (iv) trees obstructing the buildings. Most of the misclassified landscape images had strong vertical edges from tree trunks, close-ups of stems, fences, etc., that led to their assignment to the city class.



(a)

(b)

Figure 6: A subset of (a) misclassified indoor and (b) outdoor images using the color moment features; the corresponding confidence values (in %) associated with the true class are presented.

Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	94.7	95.2	83.7	83.5	94.8	95.4	95.2	95.3
Test Set	92.0	92.4	75.4	76.0	92.5	92.8	93.3	93.0
Entire Database	93.4	93.8	79.6	79.8	93.7	94.1	94.3	94.2

Table 4: Classification accuracies (in %) for city vs. landscape classification problem; the features are abbreviated as follows: edge direction histogram (EDH), edge direction coherence vector (EDCV), color histogram (CH), and color coherence vector (CCV).

We also computed the classification accuracy using the edge direction coherence vector on an independent test set of 568 outdoor images from database D1. A total of $1,177$ images of the $4,181$ outdoor images in database D1 contained close ups of human faces. We removed these images for city vs. landscape classification. Recent advances in face detection algorithms show that faces can be detected rather reliably with a high degree of accuracy [28]. Of the remaining images, we extracted 568 test images that were not part of database D2. The edge direction features yielded an accuracy of 91.5% with 49 misclassifications out of the 568 images. Combining color histogram features with the edge direction coherence vector features reduced the misclassification in the above experiment to 48, again showing that edge direction features have enough discriminative power for the city vs. landscape classification problem.

4.3.3 Further Classification of Landscape Images

A subset of 528 landscape images of database D2 were classified into the sunset, mountain, and forest classes. Of these 528 images, a human subject labeled 177, 196, and 155 images as belonging to the forest, mountain, and sunset classes, respectively. A 2-stage classifier was constructed. First, we classify an image into either sunset or the forest & mountain class. We next address the forest vs. mountain classification problem. Table 5 shows the classification results for the classification of landscape images into sunset vs. forest & mountain classes. The color coherence vector provides the best accuracy of 96.2% for the training data and 93.5% for the test data. Color features do much better than the edge direction features here, since color distributions remain more or less constant for natural images (blue sky, green grass, trees, plants, etc). A total of 26 images were misclassified (a classification accuracy of 95.1%)

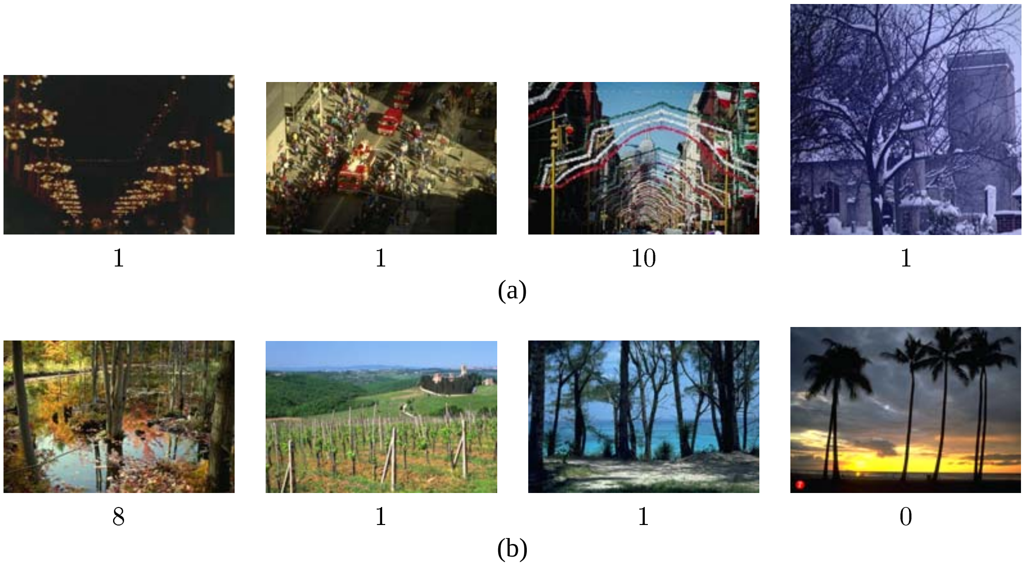


Figure 7: A subset of the (a) misclassified city images and (b) landscape images; the corresponding confidence values (in $\%$) associated with the true class using a combination of edge direction coherence vector and color histogram features.

when the color coherence vector was combined with edge direction coherence vector. We find that there is not much improvement in the classification accuracy with the combination of features. This shows that color coherence vector has sufficient discrimination ability for the given classification problem.

Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	88.3	88.3	96.2	96.2	95.9	95.9	95.5	95.9
Test Set	86.3	89.0	89.7	93.5	90.1	93.9	90.5	94.3
Entire Database	87.4	88.7	93.0	94.9	93.0	94.9	93.0	95.1

Table 5: Classification accuracies (in %) for sunset vs. forest & mountain classification.

Table 6 shows the classification results for the individual features for forest vs. mountain classes (373 images in the database). Color coherence vector provides the best accuracy of 94.7% for the training data and 91.4% for the test data. A total of 24 images were misclassified (a classification accuracy of 93.6%) when the color histogram feature was combined with either of the edge direction features. Again, the combinations of features did not perform better than the individual color features, showing that color features are quite adequate for this classification problem.

Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	83.4	78.1	92.0	94.7	94.1	92.5	93.6	93.1
Test Set	87.1	77.2	91.4	91.4	93.0	91.9	93.5	93.0
Entire Database	85.3	77.7	91.7	93.1	93.6	92.2	93.6	93.0

Table 6: Classification accuracies (in %) for forest vs. mountain classification.

4.4 Comparison with Previous Approaches

Indoor vs. outdoor classification problem was earlier addressed by Szummer and Picard [25]. The classification of outdoor images into city vs. landscape and further classification of landscape images is addressed in [12]. Both the approaches use a K-NN classifier for the respective classifications. Leave-one-out method of testing was used to measure the classification performance. A main drawback of a K-NN classifier is that all the samples need to be stored and every test image has to be compared with the samples present in the database to make a decision. Recently, there has been an added interest in using Bayesian inference for semantic content characterization in videos [29]. The approach shows how Bayesian networks can be trained for characterizing the semantic content in video, such as action, crowd, close-up, or man-made set vs. natural scenery shots. A set of low-level features is combined with hand-coded probability models to characterize the above concepts.

A Bayesian classifier generates an "optimal" classification rule under the assumed models. The VQ paradigm used to learn the class-conditional densities is similar to a neural net in its learning. Hence, the advantages and disadvantages of the Bayesian approach are similar to that of a learning algorithm using neural nets. The main advantages are:

- It provides a learning paradigm, wherein, the class-conditional feature distributions are learnt from the training samples. Thus, the entire training set need not be stored in the database, and only the learnt codebook vectors need be stored. Moreover, the number of comparisons required for a decision as compared to say a K-NN classifier, are greatly reduced (from the size of the training set to the codebook size, e.g., from $6,931$ images to 30 codebook vectors for the indoor vs. outdoor problem).

- The Bayesian paradigm not only provides a classification rule but also assigns a degree of confidence in the classification. The confidence values can be used in a reject option, wherein, results with low confidence values are rejected.

- The Bayesian paradigm allows for a simple method for the integration of multiple features through the class-conditional densities.

Our classifiers are very much similar to the Bayesian network model described in [29]. We differ in our approach for training the Bayesian classifiers. Vector Quantization is used to extract a few codebook vectors which in turn are used to estimate the class-conditional probability distributions. We further show how the MDL principle can be used to select an optimal size of codebook vectors for a given classifier and feature vector.

4.5 Discussion

Although, the Bayesian approach has a number of advantages, it has some limitations as well. The accuracy of the Bayesian approach, like any other learning paradigm, depends on two main issues:

- Feature Saliency: A feature with a low discrimination power against the pattern classes in consideration (say, indoor vs. outdoor) yields low accuracy values. For example, the edge direction and color histogram features yield accuracies of about 60% for the indoor vs. outdoor problem, yet they yield over 94% accuracy for the city vs. landscape problem. How can the feature extraction stage be automated to extract robust features?

- Training Set: More comprehensive a training set, better is the performance of the classifier. Table 7 shows the classification accuracies with increasing training set sizes on an independent test set for the indoor vs. outdoor classification problem. We can see that increasing the training set size improves the classification accuracy. When we trained the Vector Quantizer with all the available $5,081$ images using the color moment features, the system achieved a classification accuracy of around 94% on the training data (over 98% with a larger codebook size of 200 vectors). This shows that the system still has ability to learn and more the training samples, the better it can perform. A main limitation due to the above observation is that, the more biased a training set, the worse is the classifier performance over independent test sets. Is there an automatic means to select a comprehensive training set that models the underlying population well? Or can the system progressively learn over time, as new data is presented? We are currently looking at ways to add progressive/incremental learning into the classifiers.

5 Conclusion and Future Work

Content-based indexing and retrieval has emerged as an important area in computer vision and multimedia computing. User queries are typically based on semantics and not on low-level image features. It is a challenging problem to provide high-level semantic indices into large databases. In this paper, we show that certain high-level semantic categories can be learnt using low-level image features under certain constraints (test image does belong to one of the

Training Set Size	Ind. Test Set Size	Accuracy (%)
700	$2,540$	75.3
$1,418$	$2,540$	79.8
$1,768$	$2,540$	86.0
$2,192$	$2,540$	86.4
$2,541$	$2,540$	88.2

Table 7: Effect of increasing the size of training data on classification accuracies for the indoor vs. outdoor classifier; Test and training sets were different.

classes in concern), albeit the "right" set of features are used for each level of classification. Specifically, we have developed a hierarchical classifier for classifying vacation images. At the top level, vacation images are classified into indoor and outdoor categories. The outdoor images are then classified into city and landscape classes (we assume a face detector that separates close-up images of people in outdoor images into the "other" category) and finally, a subset of landscape classes are classified into the sunset, forest, and mountain class. The classification problems have been formalized using the Bayesian framework wherein Vector Quantization is used to estimate the class-conditional probability densities of the observed features. The Bayesian approach has the following advantages: (i) it not only provides a classification rule, but also assigns a degree of confidence in the classification; (ii) a small number of codebook vectors represent a particular class of images, thereby greatly reducing the number of comparisons necessary for each classification; and (iii) it naturally allows for the integration of multiple features through the class-conditional densities. Classifications based on local color moments, color histograms, color coherence vectors, edge direction histograms, and edge direction coherence vectors as features show promising results.

The accuracy of the above classifiers depends on the feature set used, the training samples, and their ability to learn from the training samples. We are currently working on adding a progressive/incremental learning paradigm to the classifiers, so that they can improve their performance over time as more training data is presented. Another challenging issue is to introduce reject option. In the simplest form, the a posteriori class probabilities can be used for rejection (rejecting images with say less than 60% probability of belonging to any class). We are looking at other means of adding the reject option into the system. Finally, we intend to add other classifiers into the system, such as day vs. night classification, people vs. non-people classification, text vs. non-text classification, etc. These classifiers can be added along with the present hierarchy to generate semantic indices into the database.

References

[1] 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*, vol. 3, pp. 231–262, 1994.

[2] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *SPIE Vol 2185: Storage and Retrieval for Image and Video Databases II*, pp. 34–47, February 1994.

[3] R. W. Picard and T. P. Minka, "Vision texture for annotation," *Multimedia Systems: Special Issue on Content-based Retrieval*, vol. 3, pp. 3–14, 1995.

[4] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing retrieval and browsing: An integrated and content-based solution," in *Proc. ACM Multimedia '95*, (San Francisco, CA), pp. 15–24, November, 5-9 1995.

[5] A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, "Virage video engine," in *Proc. SPIE: Storage and Retrieval for Image and Video Databases V*, (San Jose), pp. 188–197, February 1997.

[6] J. R. Smith and S. F. Chang, "Visualseek: A fully automated content-based image query system," in *Proc. ACM Multimedia*, pp. 87–98, Nov 1996.

[7] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. Int. Conf. on Image Proc.*, vol. 1, (Santa Barbara, CA), pp. 568–571, Oct 26-29 1997.

[8] S. Mehrotra, Y. Rui, M. Ortega, and T. S. Huang, “Supporting content-based queries over images in MARS,” in *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, 1997.

[9] H. J. Zhang and D. Zhong, “A scheme for visual feature based image indexing,” in *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, (San Jose, CA), pp. 36–46, February 1995.

[10] D. Zhong, H. J. Zhang, and S.-F. Chang, “Clustering methods for video browsing and annotation,” in *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, (San Jose, CA), February 1995.

[11] W. Y. Ma and B. S. Manjunath, “Image indexing using a texture dictionary,” in *Proc. SPIE Conference on Image Storage and Archiving System*, vol. 2606, (Philadelphia, PA), pp. 288–298, October 1995.

[12] A. Vailaya, A. K. Jain, and H. J. Zhang, “On Image Classification: City vs. Landscape,” in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, (Santa Barbara, CA), pp. 3–8, June 21 1998. To also appear in Pattern Recognition, 1998.

[13] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, “A Bayesian framework for classification of vacation images,” in *Proc. SPIE Conference Electronic Imaging ’99*, (San Jose, CA), January 1999. To appear.

[14] R. M. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, pp. 4–29, April 1984.

[15] N. Vasconcelos and A. Lippman, “Library-based coding: a representation for efficient video compression and retrieval,” in *Data Compression Conference’97*, (Snowbird, Utah), 1997.

[16] R. M. Gray and R. A. Olshen, “Vector quantization and density estimation,” in *SEQUENCES97*, 1997. http://www-isl.stanford.edu/ gray/compression.html.

[17] R. C. Dubes and A. K. Jain, “Random field models in image analysis,” *Journal of Applied Statistics*, vol. 16, no. 2, pp. 131–164, 1989.

[18] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. Tokyo: Springer-Verlag, 1995.

[19] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[20] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, “Using vector quantization for image processing,” *Proc. IEEE*, vol. 81, pp. 1326–1341, September 1993.

[21] J. Rissanen, *Stochastic Complexity in Stastistical Inquiry*. Singapore: World Scientific, 1989.

[22] M. Figueiredo and J. Leitão, “Unsupervised image restoration and edge location using compound Gauss-Markov random fields and the MDL principle,” *IEEE Transactions on Image Processing*, vol. IP-6, pp. 1089–1102, August 1997.

[23] T. Cover and J. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

[24] B. Furht, ed., *The Handbook of Multimedia Computing: Chapter* 13 - *Content-Based Image Indexing and Retrieval*. LLC: CRC Press, 1998.

[25] M. Szummer and R. W. Picard, “Indoor-outdoor image classification,” in *IEEE International Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV’98*, (Bombay, India), Jan 1998.

[26] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola, “LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms,” in *Proc. Intl’ Joint Conf. on Neural Networks*, (Baltimore), pp. I 725–730, June 1992.

[27] E. C. Yiu, “Image classification using color cues and texture orientation,” Master’s thesis, Department of EECS, MIT, 1996.

[28] H. A. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 203–208, 1996.

[29] N. Vasconcelos and A. Lippman, “A Bayesian framework for semantic content characterization,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (Santa Barbara, CA), pp. 566–571, 1998.