CLASSIFICATION OF IMAGES USING SUPPORT VECTOR MACHINES

(DESTEKÇİ VEKTÖR MAKİNESİ KULLANARAK RESİM SINIFLANDIRMA)

by

Can DEMİRKESEN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

May 2008

CLASSIFICATION OF IMAGES USING SUPPORT VECTOR MACHINES

(DESTEKÇİ VEKTÖR MAKİNESİ KULLANARAK RESİM SINIFLANDIRMA)

by

Can DEMİRKESEN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Date of Submission : April 24, 2008

Date of Defense Examination: May 8, 2008

Supervisor : Prof. Dr. Hocine CHERIFI

Committee Members : Asst. Prof. Dr. S. Murat EGI

Asst. Prof. Dr. Temel ÖNCAN

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Dr. Hocine Cherifi for his instructive comments in the supervision of the thesis. His availability, patience and generosity were very important for me to finish this work.

I am grateful to my fiancée Selin who always knew how to encourage me to continue on working in stressful moments. She helped me so much with interest and valuable comments from an engineer's point of view.

Finally I would like to thank to my family for providing me a comfortable working environment and for their support.

Can Demirkesen

May 2008

Table of Contents

List of Figures

List of Tables

Abstract

Image categorization has become more and more important in the last decade with the development of Internet, digital cameras becoming widespread and the growth in the size of image databases. Image categorization task consists of categorizing real-world natural scenes based on different features. The objective is to regroup images into semantically meaningful categories.

Computer vision researchers have been working to design computational systems that are capable of automatic scene categorization. A computational system that can perfectly mimic the human visual system and perception in order to categorize images is still missing.

In this work, the categorization task is accomplished using Support Vector Machines (SVM) that has been applied to many real-world problems producing state-of-the-art results. These include text categorization, biological data mining and handwritten character recognition. In other words SVM is a very effective method for general purpose pattern recognition and classification.

For an effective use of a classification algorithm, the data that is the subject to the classification has to be represented in a suitable way. We insisted on image representation using local, global and intermediate representations in order to obtain good results and take full advantage of SVM.

Résumé

Catégorisation d'image est devenu de plus en plus important la dernière décade avec les progresse of internet, cameras numérique et l'augmentation de taille des bases d'images. La catégorisation d'images consiste a catégoriser les scènes naturels selon différents caractéristiques d'image. Le but est de regrouper les images dans des catégories sémantiques.

Les chercheurs de vision artificielle travaillent pour concevoir un système informatique qui est capable de faire la catégorisation automatisée d'images. Un système informatique qui peut remplacer parfaitement le système visuel humain et ses capabilités de perception pour le but de catégorisation n'existe pas.

Dans ce travail, la tache de catégorisation est accompli par les Machines de Vecteur a Support (MVS) qui sont utilise dans divers application comme catégorisation de texte, data mining biologique et reconnaissance de caractère manuscrit. Autrement dit, MVS est une méthode très efficace pour la reconnaissance de modèle et pour la classification en générale.

Pour utiliser un algorithme de classification d'une manière efficace, il faut que la donnée qu'on veut classifier soit bien représentée. Nous avons insisté sur le sujet de représentation d'image en utilisant les représentations locales, globales et intermédiaires pour pouvoir bien se servir des MVS.

Özet

İnternetin gelişip büyümesi, dijital fotoğraf makinelerinin yaygınlaşması ve buna bağlı olarak resim veri tabanlarının büyümesi ile birlikte resim sınıflandırma son yıllarda büyük önem kazandı. Resim sınıflandırma esas olarak verilen bir fotoğrafın farklı özelliklere dayalı olarak sınıflandırılmasıdır. Amaç resimlerin anlamlı sınıflara ayrılmasıdır.

Yapay görme alanında çalışan araştırmacılar resim sınıflandırma işini otomatik olarak yapabilen bir sistem tasarlamaya çalışmaktadırlar. İnsanın görme ve algı yeteneklerini mükemmel bir şekilde taklit ederek resim sınıflandırabilen bir sistem henüz bulunmamaktadır.

Bu çalışmada sınıflandırma işi Destekçi Vektör Makineleri (DVM) kullanılarak yapılmıştır. DVM'ler el yazısı tanıma, doküman sınıflandırma ve veri madenciliği alanlarında kullanılan bir tekniktir. Başka bir deyişle DVM genel olarak tanıma ve sınıflandırma alanlarında kullanılan önemli bir araçtır.

Herhangi bir sınıflandırma yönteminin etkin bir şekilde kullanılması için sınıflandırılacak verinin iyi bir şekilde nitelendirilmesi zorunludur. Biz bu çalışmada resimleri niteleme konusuna özellikle önem verdik ve resimleri yerel, genel ve orta seviye özellikler olmak üzere farklı seviyelerde niteledik. Bu şekilde DVM yönteminden tam anlamıyla fayda sağlamayı amaçladık

1 INTRODUCTION

1.1 Motivation

The objective of this work is to establish an efficient system that is able to categorize pictures into semantically meaningful categories. Pictures can be regrouped in many different classes like portrait, indoor, outdoor etc. The goal of this work is to accomplish this task in real-time or in a period of time comparable to that. Such a system can offer great use in numerous areas like, large image databases, digital cameras or image classification related operations.

Image classification/categorization has become more and more important in the last decade with the development of Internet, digital cameras becoming widespread and the growth in the size of image databases. Image classification task consists of categorizing real-world natural scenes based on different features. This classification's objective is to regroup images into semantically meaningful categories.

Because of the difficulty of the problem, a combined cognitive and computational approach is followed to understand and implement scene categorization. Computer vision researchers have been working to design computational systems that are capable of automatic scene categorization.

In this work, the classification task is accomplished using Support Vector Machines that have been applied to many real-world problems producing state-of-the-art results. These include text categorization, image classification, biosequence analysis, biological data mining, engine knock detection, database marketing and handwritten character recognition.

1.2 Thesis Organization

This thesis is organized as follows: in Introduction (Chapter 1) the subject and the scope of this thesis is presented with brief presentation of tools and techniques which are needed to be used. In Related Work (Chapter 2) a detailed survey of literature is presented. And the main algorithms and techniques are studied in detail. In Image Representation Chapter (Chapter3), local, global and intermediate image representation approaches are studied and detailed review of the literature related in these subjects are presented as well. In Image Compression Chapter (Chapter 4) state-of-the-art image compression techniques are reviewed. In Experimentations Chapter (Chapter 5) a series of experimentations about image representation, image compression, and image classification are presented and the results are discussed. Finally in Conclusions Chapter (Chapter 6) the essential ideas about this work are summarized.

2 RELATED WORK

2.1 Introduction

In this chapter a detailed survey of literature of image categorization area is presented. And the main algorithms and techniques are studied in detail. Support Vector Machines are specifically presented in detail. Multi-class classification strategies using SVM are studied as well. This chapter has an important role for pointing out the reasons why we have chosen SVM as a classification tool for our image classification objective.

2.2 Previous Work on Image Categorization

The objective of the classification or pattern recognition task is to optimally extract patterns based on certain conditions and to separate one class from the others. Pattern recognition was often achieved using linear and quadratic discriminants, the k-nearest neighbor classifier, template matching and Neural Networks. These methods are basically statistic. The problem of using these recognition methods has to construct a classification rule without having any idea of the distribution of the measurements in different groups. Support Vector Machines (SVMs) have gained prominence in the field of pattern classification. They are competing with other techniques such as template matching and Neural Networks for pattern recognition.

The previous works on the image classification subject can be studied by the classification approaches that have been preferred. We can consider different categories that correspond to different classification techniques while reviewing these previous works.

The construction of a classification procedure from a set of data for which the true classes are known has been variously termed pattern recognition, discrimination, or supervised learning (in order to distinguish it from unsupervised learning or clustering in which the classes are inferred from the data). Classification is studied in two basic categories which are supervised classification and unsupervised classification. If a labeled set of data points are available, supervised classification is applied. In the contrary case unsupervised classification is performed. In the scope of this work, supervised classification will be used. A classifier is a system that performs a mapping from a feature space X to a set of labels Y. Basically what a classifier does is to assign a pre-defined class label to a sample. This should not be confused with clustering where the algorithm autonomously partitions the data into clusters in a way so that the data in each cluster is grouped in feature space. Supervised classification is a method where you decide the classes while clustering is an unsupervised method where the algorithm groups the data automatically.

Three main historical subjects of research can be identified: statistical, machine learning and neural network. These have largely involved different professional and academic groups, and emphasized different issues. All groups have however had some objectives in common. They have all attempted to derive procedures that would be able:

- to equal, if not exceed, a human decision-maker's behavior, but have the advantage of consistency and, to a variable extent, explicitness,
- to handle a wide variety of problems and, given enough data, to be extremely general,
- to be used in practical settings with proven success.

The following is a brief description of each approach. Let *c1,c2,...,cn* be the finite set of n classes for an image scene. The probability *P(c,f)* gives the likelihood that the correct class is *c,,* for the d-dimensional feature vector f. There are two issues to be considered. The first is the a *priori* probability of each class. Fortunately, this issue is not a critical matter since it can be estimated from the design data set or it can be assumed to be equal for all classes. The second and major issue is to estimate the class conditional

probability *P(c/f),* for each class. Towards that goal, two main directions are usually considered parametric and non-parametric estimation.

2.3 Parametric Methods

In parametric methods we consider the ideal case in which the probability structure underlying the categories is known perfectly. This sort of situation does not occur frequently in real problems

2.3.1 Parametric Bayes Classifier

In this approach, an a priori form of the class conditional density p(c|fi) where i={1,…,n} is assumed; the parameters in this density are to be estimated. The design data are used to estimate these parameters. In the Gaussian case the only parameters needed to describe P(ci|f) are the covariance matrix and the mean vector. There are many approaches in the literature to estimate these two parameters. One of the most common approaches is the maximum likelihood estimator.

2.3.2 Fisher's Linear Discriminant

This is one of the oldest classification procedures, and is the most commonly implemented in computer packages. The idea is to divide sample space by a series of lines in two dimensions, planes in 3-D and, generally hyper planes in many dimensions. The line dividing two classes is drawn to bisect the line joining the centers of those classes; the direction of the line is determined by the shape of the clusters of points [1].

2.3.3 Naive Bayes Classifier

A naive Bayes classifier is a basic probabilistic classifier based on the Bayes Theorem with strong independence assumptions. Naïve Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood which is a statistical method used to calculate the best way of fitting a mathematical model to

some data. Modeling real world data by estimating maximum likelihood offers a way of tuning the free parameters of the model to provide an optimum fit.

Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. Recently, careful analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers [2]. An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix

2.3.4 The Naive Bayes Probabilistic Model

Abstractly, the probability model for a classifier is a conditional model $p(C|F_1,...,F_n)$ over a dependent class variable C with a small number of outcomes or *classes*, conditional on several feature variables F_1 through F_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. Therefore the model is reformulated to make it more tractable.

Using Bayes Theorem, we write:

$$p(C|F_1,...,F_n) = \frac{p(C)p(F_1,...,F_n|C)}{p(F_1,...,F_n)} \tag{2.1}$$

In plain English the above equation can be written as:

$$posterior = \frac{prior \times likelihood}{evidence} \tag{2.2}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint

probability model $p(C, F_1, ..., F_n)$ which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$p(C, F_1, ..., F_n)$$

$$= p(C)p(F_1, ..., F_n|C)$$

$$= p(C)p(F_1|C)p(F_2, ..., F_n|C, F_1)$$

$$= p(C)p(F_1|C)p(F_2|C, F_1)p(F_3, ..., F_n|C, F_1F_2) \quad (2.3)$$

and so forth. Now with the naive conditional independence assumptions: each feature F_i is conditionally independent of every other feature F_j for i≠j. This means that

$$p(F_i|C, F_j) = p(F_i|C) \quad (2.4)$$

and so the joint model can be expressed as:

$$p(C, F_1, ..., F_n) = p(C)p(F_1|C)p(F_2|C)p(F_3|C)...$$

$$= p(C)\prod_{i=1}^{n} p(F_i|C) \quad (2.5)$$

This means that under the above independence assumptions, the conditional distribution over the class variable *C* can be expressed like this:

$$p(C|F_1, ..., F_n) = \frac{1}{Z} p(C)\prod_{i=1}^{n} p(F_i|C) \quad (2.6)$$

where *Z* is a scaling factor dependent only on $F_1, ..., F_n$, i.e., a constant if the values of the feature variables are known.

Models of this form are much more manageable, since they factor into a so-called *class prior* $p(C)$ and independent probability distributions $p(F_i|C)$. If there are *k* classes and if

a model for $p(F_i)$ can be expressed in terms of r parameters, then the corresponding naive Bayes model has $(k - 1) + n\ r\ k$ parameters. In practice, often $k = 2$ (binary classification) and $r = 1$ (Bernouilli variables as features) are common, and so the total number of parameters of the naive Bayes model is $2n + 1$, where n is the number of binary features used for prediction.

Vailaya et al. [3] designed binary Bayesian classifiers for hierarchical classification of vacation images based on color and texture representations. For this study, every image belongs to one category. At the highest level, images are classified as indoor or outdoor; outdoor images are further classified into city or landscape; finally, a subset of landscape images is classified into sunset, forest, and mountain classes, which have 90.5%, 95.3%, and 96.6% classification accuracies, respectively.

2.3.5 Hidden Markov Model

A hidden Markov model (HMM) is a tool for representing probability distributions over sequences of observations. Let the variable Y_t denote the observation at time t. This can be a symbol from a discrete alphabet, a real-valued variable, an integer, or any other object, as long as a probability distribution can be defined over it. The HHM gets its name from two defining properties. First, it assumes that the observation at time t was generated by some process whose state St is hidden from the observer. Second, it assumes that the state of this hidden process satisfies the Markov property: that is, given the value of S_{t-1}, the current state S_t is independent of all the states prior yo t-1. In other words, the state at some time encapsulates all we need to know about the history of the process in order to predict the future of the process. The outputs also satisfy a Markov property with respect to the states: S_t, Y_t is independent of the states and observations at all other time indices. Statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. A HMM can be considered as the simples Bayesian network.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the

state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.
Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, musical score following, partial discharges and bioinformatics [4].

2.3.6 Architecture of a Hidden Markov Model

The diagram below shows the general architecture of an instantiated HMM (Figure 2.1). Each oval shape represents a random variable that can adopt a number of values. The random variable $x(t)$ is the hidden state at time t (with the model from the above diagram, $x(t) \in \{x_1, x_2, x_3\}$). The random variable $y(t)$ is the observation at time t ($y(t) \in \{y_1, y_2, y_3, y_4\}$). The arrows in the diagram (often called a trellis diagram) denote conditional dependencies.

From the diagram, it is clear that the value of the hidden variable $x(t)$ (at time t) *only* depends on the value of the hidden variable $x(t-1)$ (at time $t-1$). This is called the Markov property. Similarly, the value of the observed variable $y(t)$ only depends on the value of the hidden variable $x(t)$ (both at time t).

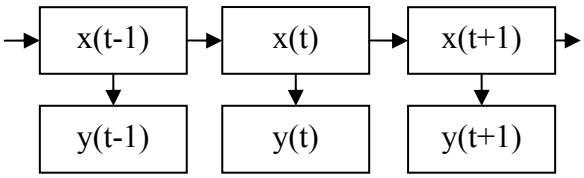

Figure 2.1 HMM Diagram

Li and Wang [5] proposed an automatic linguistic indexing of pictures (ALIP) system that uses the 2D multi-resolution hidden Markov model on features of image. They extracted color components of pixels in 4x4 blocks and energy in high-frequency bands

of wavelet transforms for texture. Color features are extracted using LUV color space. They applied either Daubechies-4 or Haar transform for extracting texture features.

2.4 Non Parametric Methods

Non parametric methods move yet further from the Bayesian ideal, and we assume that there is no prior parameterized knowledge about the underlying probability structure; in essence our classification will be based on information provided by training samples alone. Classical techniques such as the nearest-neighbor algorithm play an important role here.

2.4.1 Non-Parametric Estimation using Parzen Window

In the non-parametric approach, no a priori structural form is assumed for P(ci|f). The Parzen window approach estimates P(ci|f) for any x using the number of samples in a hypercube of dimension h around f. Mahmoud and El-Melegy used this method for remote-sensed image classification [6].

2.4.2 Non-Parametric Estimation using k-Nearest Neighbor

The approach directly estimates the a posteriori probabilities. The k-Nearest neighbor rule classifies a point f by assigning it to the class that is most frequently represented among the k nearest samples.

The k-nearest neighbour algorithm is a supervised classification technique. It functions on the intuitive idea that close objects are more likely to be in the same category. The class label of a new instance is found using the majority vote of its k-nearest neighbours. K is a small positive integer number. If k=1 the new instance is assigned to the same class with its nearest neighbour. In binary classification problems, it is helpful to choose k to be an odd number in order to avoid difficulties that can be caused by the same number of vote. For example if k=4 and if two of these neighbours belong to class A, and the other two neighbours belong to the class B, the algorithm cannot decide whether the new instance belong to the class A or class B. The neighbors are taken from a set of instances for which the correct class label is known. One can consider

these neighbors as a training set for the algorithm, though no explicit training step is required. *k*-Nearest Neighbors is a memory-based method that, in contrast to other statistical methods, requires no training (i.e., no model to fit). In order to identify neighbors, the instances are represented by position vectors in a multidimensional feature space. A number of distance measures can be used such as the Euclidian distance or the Manhattan distance. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data. The prototype examples are vectors in a multidimensional feature space. The 'training' phase of the algorithm consists only of storing the feature vectors and class labels of the prototype samples. In the classification phase, the test sample (whose class is not known) is represented as a vector in the feature space. Distances from the new vector to all previously stored prototype vectors are computed and *k* closest samples are selected. There are a number of ways to classify the new vector to a particular class; one of the most used techniques is to predict the new vector to the most common class amongst the K nearest neighbors. A major drawback to use this technique to classify a new vector to a class is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the K nearest neighbors when the neighbors are computed due to their large number. One of the ways to overcome this problem is to take into account the distance of each K nearest neighbors with the new vector that is to be classified and predict the class of the new vector based on these distances [1].

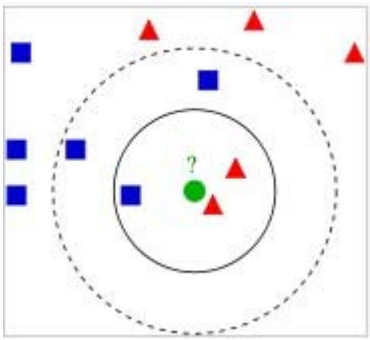

Figure 2.2 Example of k-NN classification

An example of *k*-NN classification is illustrated in the figure 2.2. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ it is classified to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ it is classified to first class (3 squares vs. 2 triangles inside the outer circle) [2]. Vogel and Schiele [7] proposed a semantic typicality measure using k-nearest neighbor to classify sub regions. They represent images by concept occurrence vectors consisting of the frequencies of local semantic concepts like sky, water, grass, trunks, foliage etc. They did the classification task measuring the typicality by computing the Mahalanobis distance between the images. The features they used to represent image sub regions are 84-bin HIS color histogram and 72-bin edge direction histogram.

2.4.3 Artificial Neural Networks

Artificial neural network (ANN) is a powerful tool widely used in the context of classification. There is an important number of work in image classification where ANNs are used as a classification technique. ANN is basically a mathematical structure based on biological neural networks. This structure consists of interconnected artificial neurons which process information. In the learning phase, ANNs can be adapted to a problem by changing its structure based on the information flow through the network. ANNs are used to model complex relationships between inputs and outputs or to recognize patterns in the flow of data. An artificial neural network can be considered as a function f(x) ANNs have the network characteristic because of the function f(x) is defined as a composition of other functions which can further be defined as a composition of other functions.

2.4.4 Artificial Neuron

An artificial neuron is a mathematical representation of a biological neuron. It has multiple input and output that correspond respectively to dendrites and axons of the biological one. Actions of biological neurons are simulated with numerical coefficients applied to inputs. In a mathematical point view, an artificial neuron is a real valued function with multiple variables. Consider the general case of an artificial neuron with m inputs ($x_1,\ldots,x_m$), an artificial neuron model is a calculation rule that associates an

output value to these input. It is effectively a function with m variables. In the model of McCulloch and Pitts a synaptic weight w_i is associated to each input. The first operation of the neuron is to calculate the weighted sum of the input values in the following way:

$$w_1 x_1 +,...,+w_m x_m = \sum_{j=1}^{m} w_j x_j \tag{2.7}$$

A w_0 is added to this, and the result is transformed with nonlinear activation function φ called transfer function. The output corresponding to the input $x_1,\ldots x_m$ is then obtained:

$$\varphi\left(w_0 + \sum_{j=1}^{m} w_j x_j \right) \tag{2.8}$$

Which is simplified adding a fictive input $x_0=1$:

$$\varphi\left(\sum_{j=0}^{m} w_j x_j \right) \tag{2.9}$$

In the original formulation of McCulloch and Pitts, Heaviside function is used as the activation function which has 0 or 1 as output value. In this case the output is:

$$\varphi\left(\sum_{j=1}^{m} w_j x_j - w_0 \right) \tag{2.10}$$

If the sum

$$\sum_{j=1}^{m} w_j x_j > w_0 \tag{2.11}$$

Then the output is 1. Else it is 0. w_0 is then the activation threshold of the neuron. The output 0 corresponds to a inactive neuron [8].

2.4.5 Support Vector Machines

Support vector machines are a core machine learning technology. They have strong theoretical foundations and excellent empirical successes. We shall consider SVMs in the binary classification setting. SVM can be applied to multi-class problems as well using "one against one" approach combining several binary classifiers [9] or "one-against-all approach" training m SVM classifiers where each classifier distinguishes images in one category from all other m–1 categories [10].

We are given the training data {x1 ... xn} that are vectors in some space X and their labels {y1 … yn} where y_i in {-1, l}. In their simplest form, SVMs are hyper planes that separate the training data by a maximal margin. All vectors lying on one side of the hyper plane are labeled as -1, and all vectors lying on the other side are labelled as 1. The training instances that lie closest to the hyper plane are called support vectors. If the data are linearly non-separable but non-linearly separable, the non-linear SVM classifier will be applied [11, 12].

The basic idea is to transform input vectors into a high dimensional feature space using non-linear transformation Φ, and then to do a linear separation in feature space. To construct a non-linear SVM classifier, inner product < x, y > is replaced by a kernel function K(x, y).

A kernel is a function K(x, y) that given two vectors in input space, returns the dot product of their images in feature space [11].

$$k(x, y) = \langle \Phi(x).\Phi(y) \rangle \tag{2.12}$$

There are several different kernels; choosing one depends on the task at hand. One of the simplest is the polynomial kernel

$$k(x, y) = \langle x.y \rangle^d \tag{2.13}$$

For example, taking d = 2 and x, y in RxR

$$\begin{aligned}\langle x.y\rangle^2 &= (x_1y_1 + x_2y_2)^2 \\ &= (x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2) \\ &= (x_1^2, x_2^2, \sqrt{2}x_1x_2)(y_1^2, y_2^2, \sqrt{2}y_1y_2) \\ &= \langle \Phi(x).\Phi(y)\rangle \end{aligned} \tag{2.14}$$

Defining:

$$\Phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \tag{2.15}$$

The dual representation of the decision function is:

$$f(x) = sign\left(\sum_{i=1}^{l} y_i\alpha_i < x.x_i > +b\right) \tag{2.16}$$

If the decision function is considered for the optimal hyper plane classifier in dual form and apply the mapping Φ to each vector it uses, we obtain:

$$f(x) = sign\left(\sum_{i=1}^{l} y_i\alpha_i < \Phi(x).\Phi(x_i) > +b\right) \tag{2.17}$$

Then the kernel function is applied which will provide a non-linear decision function of the form:

$$f(x) = sign\left(\sum_{i=1}^{l} y_i\alpha_i k(x, x_i) + b\right) \tag{2.18}$$

The function f is used to classify a new point x.

2.4.6 Multi-class SVMs

The problem of multi-classification, especially for systems like SVMs, does not present an easy solution. It is generally simpler to construct classifier theory and algorithms for two mutually exclusive classes than it is for N mutually exclusive classes. Platt [13] claimed that constructing N-class SVMs is still an unsolved research problem. Definition of the k-class classification problem is as follows:

- Given *l* independent and identically distributed sample *(x1,y1)…(xl,yl)* where x_i for *i=1…l,* is a feature vector and y_i is the class label of the corresponding x
- Find a classifier with the decision function, $f(x)$ such that $y = f(x)$, where y is the class label for x

Support vector machine is basically a binary classifier however there exist several methods that can be applied to the original SVM algorithm to deal with k-class classification problem. It is possible to extend binary SVMs to multi-class problems by combining them or modify the original algorithm without combining any SVM. These different approaches can be categorized in two major titles. The first one is called 'all-in-one' where all the data is considered in one optimization formulation. This approach consists of modifying the original algorithm. The second one is called 'divide-and-combine' which consists of dividing the multi-class problem into several sub-problems that can be solved by binary SVMs.

The methods which have been proposed for solving the k-class problem are listed below:

- Divide-and-combine
  - Using *k* one-against-all classifiers.
  - Using $k(k-1)/2$ one-against-one classifiers with one of the voting scheme listed below:
    - Majority Voting
    - Pairwise Coupling
  - Decision Directed Acyclic Graph (DDAG)
  - Divide-By-2 (DB2)

- All-in-one
  - Construct the decision function by considering all classes at once.
  - Construct a decision function for each class by only considering the training data points belong to that particular class.

2.4.7 Divide-and-combine Methods

The multi-class classification problem refers to assigning each of the observations into one of k-classes. As two-class problems are much easier to solve, many authors propose to use two-class classifiers for multi-class classification. The methods described in the following are used to transform multi-class problems to two-class problems.

2.4.7.1 One-against-all

This is the simplest scheme. *K* classifiers will be constructed, one for each class. The ith classifier will be trained to classify the training data of class *i* against all other training data. The decision function for each of the classifier will be combined to give the final classification decision on the *K*-class classification problem. Mathematically the ith SVM solves the following problem that yields the ith decision function:

$$f_i(x) = w_i^T \Phi(x) + b_i \tag{2.19}$$

$$\min L(w_i, \xi_j^i) = \frac{1}{2}\|w_i\|^2 + C\sum_{j=1}^{N} \xi_j^i \tag{2.19a}$$

$$\text{Subject to } y_j\left(w_i^T \phi(x_j) + b_i\right) \geq 1 - \xi_j^i, \xi_j^i \geq 0, \tag{2.19b}$$

Where y_j = 1 if y_j = i and y_j =-1 otherwise.

At the classification phase, a sample x is classified as in class i* for which *f*i* produces the largest value

$$\text{I*} = \underset{1\ldots K}{\arg\max}\, f_i(x) = \underset{1\ldots K}{\arg\max}\, (w_i^T \Phi(x) + b_i) \tag{2.20}$$

For each point, we have *K* SVM decision outputs $f_k(x), 1 \le k \le K$. The class of a point is given as $\arg\max_k f_k(x)$. The final output is the class that corresponds to the SVM with the highest output value. The disadvantage of this method is that the number of training samples is too large, so it is difficult to train.

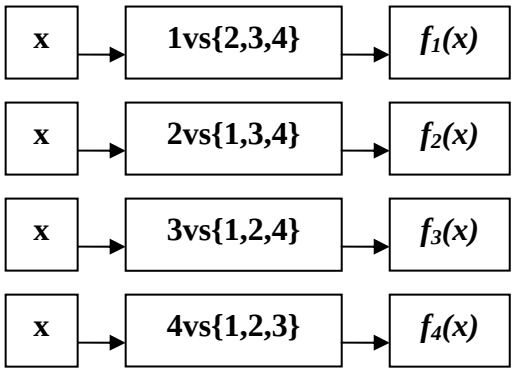

Figure 2.3 Four one-against-all classifiers.

The instances of the class labeled as 1 are separated from the other instances to build the first classifier shown in Figure 2.3. Xuchan Li et al. [14] classified artificial and real world images, they represented images by feature vectors of color moments consisting of the mean, variance and skewness. They used multi-label SVM, one-versus-all method to combine predictions of multiple binary SVM classifiers. Chen and Wang [15] have a region-based approach for image representation. They described regions by color, texture and shape attributes. For color they used LUV color space and for texture they used square root of the second order moment in high frequency bands. Tsai et al. [16] chose to represent images with HSV and 3 level Daubechies-4 wavelet decompositions which contain texture information. They have region based approach and they used SVM with one-against-all method for classification. Williamowski et al. [17] preferred to work with Harris affine detector and SIFT descriptor to represent images and they compared Naïve Bayes and SVM classifiers based on these features. Fan et al. [18] proposed concept sensitive salient objects to enable more expressive representation of image contents that can be obtained by wavelet transformation.

Chapelle et al. [19] restricted themselves to global and low-level features; they use HSV color histogram (16 bins per color) and chose SVM for classification purpose.

2.4.7.2 One-against-one

Another major method is called the one-against-one method. It is also called the pairwise coupling. Pairwise coupling is a popular multi-class classification method that combines all comparisons for each pair of classes. The total number of classifiers for a *K*-class problem will then be $K(K\text{-}1)/2$ where each one is trained on data from two classes. The training data for each classifier is a subset of the available training data, and it will only contain the data for the two involved classes. The data will be reliable, i.e. one will be labeled as +1 while the other as -1. For training data from ith and jth classes, the following binary classification problem is solved:

$$\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}\left(w^{ij}\right)^T w^{ij} + C\sum_t \xi_t^{ij} \tag{2.21}$$

There are different methods for doing the future testing after all the k(k–1)/2 classifiers are constructed. A common way to combine pairwise comparisons is by voting. It constructs a rule for discriminating between every pair of classes and then selecting the class with the most winning two-class decisions. Though the voting procedure requires just pairwise decisions, it only predicts a class label. Each of the $K(K-1)/2$ binary SVM classifiers provides a partial decision for classifying a point. The study of [33] shows that combining these decision outputs differently may yield different class decisions. To combine these classifiers, it naturally adopts Max Wins Algorithm that finds the resultant class by first voting the classes according to the results of each classifier and then choosing the class that is voted most. Friedman shows circumstances in which this algorithm is Bayes optimal. Kreßel applies the Max Wins algorithm to Support Vector Machines with excellent results [20]. The disadvantage of this method is that the number of classifiers is too many for every two classes need to be compared. So the time of testing is slow. In many scenarios, probability estimates are desired beside the class label.

A binary classifier decides whether a point x belongs to class w_i or w_j. The probability that x belongs to class w_i, given that x is in either class w_i or w_j, can be written as:

$$p_{ij} = P\left(x \in w_i \mid x, x \in w_i \cup w_j\right) \quad (2.22)$$

With pij, we can calculate the estimate pi of the a posteriori probability $p\left(x \in w_i\right)$ by using a matrix of p_{ij} and $p_{ji} = 1-p_{ij}$ to compute pi as follows**:**

$$p_i = \frac{2}{K(K-1)} \sum_{j \neq i} p_{ij} \quad (2.23)$$

and the classification is given by:

$$\arg\max_{1 \leq i \leq k} p_i \quad (2.24)$$

Where

$$p_{ij} = \frac{1}{2} f_{ij}\left(x\right) + 0.5 \quad (2.25)$$

Using the following functions:

$$\sigma\left(p_{ij}\right) = \begin{cases} 1, p_{ij} \geq 0.5 \\ 0, otherwise \end{cases} \quad (2.26)$$

$$\sigma\left(p_{ij}\right) = p_{ij} \quad (2.27)$$

$$\sigma\left(p_{ij}\right) = \frac{1}{1 + e^{-1/2\, p_{ij} - 0.5}} \quad (2.28)$$

$$\sigma(p_{ij}) = \begin{cases} 1, if p_{ij} \geq 0.5 \\ x, otherwise \end{cases} \tag{2.29}$$

We can write the following decision function:

$$p_i = \frac{2}{K(K-1)} \sum_{j \neq i} \sigma(p_{ij}) \tag{2.30}$$

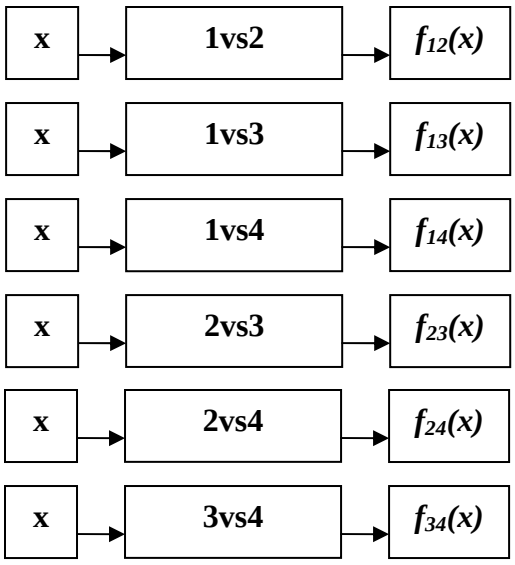

Figure 2.4 Six one-against-one classifiers

In order to build the six classifiers (in Figure 2.4) the training examples are taken one versus one for each couple (i,j) i≠j. Gao and Fan [21] represented images with the following features: 3-dimensional R,G,B average color; 4-dimensional R,G,B color variance; 3-dimensional L,U,V average color; 4-dimensional L,U,V color variance; 2-dimensional average & standard deviation of Gabor filter bank channel energy; 30-dimensional Gabor average channel energy; 30-dimensional Gabor channel energy deviation; 2-dimensional Coarse & Contrast Tamura texture feature and 5-dimensional angle histogram derived from Tamura texture. They decompose the multi class problem into a set of one-against-one binary problems. Class labels used in their experiments are

as follows: battle, plane, elephant, remnant, leaf, cloud, ship, purple, flower, brown, horse, tree, sail, cloth, sky, snow, rock, red flower, sand field, yellow flower, grass, sea, water. Their method yields a mean accuracy 87.07% with 7.28% standard deviation across all binary problems with the range as (67.64%, 99.44%). Zhu et al. [22] used frequency distribution, edge orientation and color histograms as global features and they incorporated these features with embedded text lines to improve the image classification accuracy.

2.4.7.3 Decision Directed Acyclic Graph (DDAG)

The Decision Directed Acyclic Graph (DDAG) is used to combine many two-class classifiers into a multi-class classifier. For an N-class problem, the DDAG contains N(N-1)/2 classifiers, one for each pair of classes. The DDAG contains N(N-1)/2 nodes, each with an associated 1-v-1 classifier. The algorithm designed for multi-class classification based on placing 1-v-1 SVMs into nodes of a DDAG is called DAGSVM, it is efficient to train and evaluate. A Directed Acyclic Graph (DAG) is a graph whose edges have an orientation and no cycles. A Rooted DAG has a unique node such that it is the only node which has no arcs pointing into it. A Rooted Binary DAG has nodes which have either 0 or 2 arcs leaving them. We will use Rooted Binary DAGs in order to define a class of functions to be used in classification tasks. The class of functions computed by Rooted Binary DAGs is formally defined as follows: given a space X and a set of Boolean functions $F=\{f: X \rightarrow \{0.1\}\}$, the class DDAG(F) of Decision DAGs on N classes over F are functions which can be implemented using a rooted binary DAG with N leaves labeled by the classes where each of the K=N(N-1)/2 internal nodes is labeled with an element of F. The nodes are arranged in a triangle with the single root node at the top, two nodes in the second layer and so on until the final layer of N leaves. The i-th node in layer j<N is connected to the j-th and (i+1)-st node in the (j+1)-st layer.

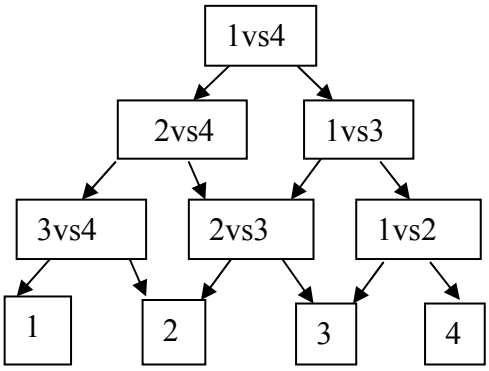

Figure 2.5 The decision DAG for finding the best class out of four classes.

To evaluate a particular DDAG G on input x, starting at the root node, the binary function at a node is evaluated. The node is then exited via the left edge, if the binary function is zero; or the right edge, if the binary function is one. The next node's binary function is then evaluated. The value of the decision function D(x) is the value associated with the final leaf node (see Figure 2.5). The path taken through the DDAG is known as the *evaluation path*. The input x reaches a node of the graph, if that node is on the evaluation path for x. We refer to the decision node distinguishing classes *i* and *j* as the *ij*-node. Assuming that the number of a leaf is its class, this node is the i-th node in the (N-j+i)-th layer provided i<j. Similarly the j-nodes are those nodes involving class j that is, the internal nodes on the two diagonals containing the leaf labeled by j.

For the DAGSVM, the choice of the class order in the list (or DDAG) is arbitrary. We simply use a list of classes in the natural numerical (or alphabetical) order. Limited experimentation with re-ordering the list did not yield significant changes in accuracy performance.

2.4.7.4 Divide-By-2 (DB2)

Starting from the whole data set, DB2 hierarchically divides the data into two subsets until every subset consists of only one class. DB2 divides the data such that instances belonging to the same class are always grouped together in the same subset. Thus, DB2 requires only $N-1$ classifiers. The basic strategy is to divide the data into two subsets at every hierarchical level. To group the N classes into two, different criteria can be

used. The division step can be considered as a clustering problem. One method is to use k-means clustering. An even simpler method is to divide them based on their class mean distances from the origin. Figure 2.6 illustrates the algorithm flow of the training process for a five class data sample.

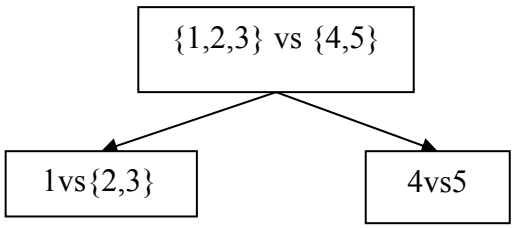

Figure 2.6 An example training phase of DB2 for 5 classes

The training phase can be summarized as follows:

1. Divide all the data samples into two subsets, A and B.
2. Apply SVM to A and B and find the parameters of the decision boundary separating them.
3. Repeat the steps for both A and B until all the subclasses include only one class.

DB2 training leads to a binary decision tree structure for testing.

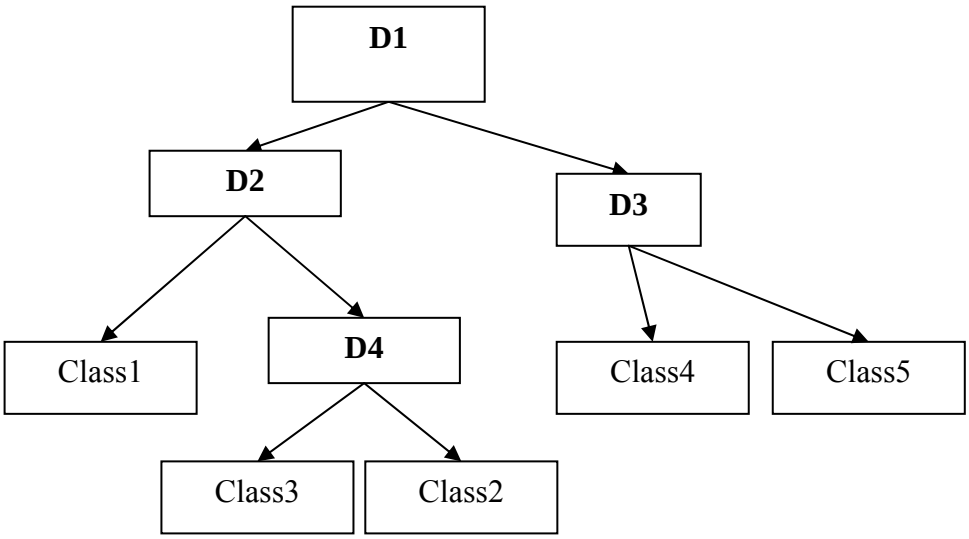

Figure 2.7 DB2 decision tree

Figure 2.7 illustrates the decision tree that we built for the testing phase of the five class problem. At the beginning, all the classes are assumed to be nominees of the true class. At every node, after applying the corresponding decision function to the test input, the nominees that do not exist in the region (positive or negative) in which the test input belongs, are eliminated. Following the branches that indicate the same labels as the result of the decisions, we end up with the predicted class. The best case occurs if we find the predicted class at the first node, and the worst case occurs if we find the predicted class after applying all the $N - 1$ decision functions. In one-against-one, a test data is applied to all $N \times (N - 1)/2$ classifiers. For one against-rest exactly N classifiers and for DAGSVM exactly $N - 1$ classifiers are applied.

2.4.8 All-in-one Method

It has been observed that there are certain limitations of the approaches that extend binary SVMs to multi-class problems. One of these limitations is that they do not consider the full problem directly. Particularly, the one-against-all approach degrade the balance of the training sets (there are far more negative training examples in each binary classifier's training set), and the one-against-one method uses only information from the two classes that it works with. Each one-against-one classifier loses the information from all the remaining classes. On the other hand, there exists a more natural approach that considers the multi-class problem directly as a generalization of the binary classification algorithm [23, 24]. This more natural way to solve k-class problems is to construct a piecewise linear separation of the k classes in a single optimization. In the first approach, one decision function will be constructed for each class (similar to the K 1-to-rest classifiers method). The optimization problem in the SVM formulation can be generalized to consider all the decision functions at once. The constraint is also relaxed. Instead of enforcing all decision functions to give zero value at the decision boundary, it is sufficient that the output of the decision function for the correct class is greater than the rest of the decision function by a margin of 2. The binary SVM optimization problem is generalized to the following:

Minimize

$$\min \phi(w,\xi) = \frac{1}{2}\sum_{m=1}^{k}(w_m \cdot w_m) + c\sum_{i=1}^{l}\sum_{m\neq y_i}\xi_i^m \tag{2.31}$$

Subject to

$$\left(w_{y_i} \cdot x_i\right) + b_{y_i} \geq \left(w_m \cdot x_i\right) + b_m + 2 - \xi_i^m,\ \ \xi_i^m \geq 0, i = 1,...,l\ \ m \in \{1,...,k\} \setminus y_i \tag{2.31a}$$

This gives the decision function:

$$f(x) = \arg\max_n \left[\left(w_n \cdot x\right) + b_n\right], n = 1,...;k. \tag{2.32}$$

This approach is very interesting because it considers all the examples and classes at the same time, without losing any important information for finding to the best solution for each problem. Besides, the SVM obtained that way does not need as much support vectors as the others do [25] and performs better in the cases where the training set is separable. If not, the misclassified examples can be penalized leading to solutions that are biased towards these examples.

The difference between binary classification and k-class problem is that in the binary classification case the labels yi belonged to {-1, +1}, which now belong to {1, 2, …, k} in k-class problem. Therefore, the binary SVM can be generalized to multi-class by using a different weight vector and bias for each class (w_j and b_j for j E {1,2,. . . , ***k}).*** So, this classifier computes ***k*** outputs to classify any pattern. The classification function is then:

$$f(x) = \arg\max_{j\in\{1,...,k\}} \left(\phi^T(x)w^j + b^j\right) \tag{2.33}$$

Where x is the vector to be classified.

Barla et al. [26] used HSV color histograms and co-occurrence matrices to built SVM classifiers. They worked on indoor-outdoor images and cityscapes. Some other

techniques are used in image classification: Guan et al. [27] proposed an automatic statistical approach to categorize traditional Chinese paintings in three classes: figure painting, landscapes and flower-and-bird paintings. They integrated texture and color information used Gabor wavelet to extract texture information and color histogram for color (they work on 8x8x8 RGB color space). They used relative-distance based voting rule to categorize images. Oliva and Torralba [28] proposed a technique which is based on a very low dimensional representation of the scene, that they term the Spatial Envelope. They proposed a set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) that represent the dominant spatial structure of a scene. The model generates a multidimensional space in which scenes sharing membership in semantic categories (e.g. streets, highways, and coasts) are projected closed together. Barnard and Forsyth [29] applied a hierarchical statistic model to generate keywords for classification based on semantically meaningful regions.

The goal of classifier ensembles and multiple classifier systems is to improve the classification accuracy of a single classifier by using the results of many. Szummer and Picard [30] use *k*-nearest neighbor and SVM classifiers, respectively, for separately classifying color and texture features into indoor and outdoor classes, and then design a combiner to *vote* or the output classes produced by the first-level classifiers so as to make the final decision. They show that their classifiers have 90.3% and 90.2% classification accuracies, respectively.

2.5 Conclusion

This chapter has been very useful for further work. We have seen all the tools that can be used for our image categorization objective. It is very important to know what other researchers have done before starting to work. We have seen in the literature survey how researcher used different classification techniques other then SVM, and how SVM is used in different works on image categorization or other categorization problems. It has been very important to study different approaches for the use of SVM as a multi-class classification method. There is a certain number of combination approaches for SVM to build multi-class classifiers which are all presented in detail with examples.

3 IMAGE REPRESENTATION

3.1 Introduction

This chapter is dedicated to image representation subject. Image representation has an essential role for image processing and for image classification. Now that the objective of this thesis is make some progress in image classification, one of the focus of this work has to be the image representation issue. It will be shown in this chapter that for image representation there is three different approaches namely global, local and intermediate image representation each of which will be discussed in detail.

3.2 Local and Global Features

Having an automatic system categorize a scenery image is not a trivial task. A good understanding of human categorization capability is required in order to simulate it on a computer system. Therefore automatic scene categorization is closely related to cognitive sciences.

Early research (Barrow and Tannenbaum, 1978 [31]; Marr, 1982 [32]) in this area, have described the scene recognition task as a progressive reconstruction of local measurements successively to integrate those into decision layers in order to decide the category of scene. In contrast, some experimental studies have suggested that recognition of real world scenes may be initiated from the global configuration (like spatial layout), ignoring most of the details and object information (Biederman, 1988 [33]; Potter, 1976 [34]). In their work, Renniger and Malik discuss the same matter as

Oliva and Torralba do in their work [28]. In the latter the authors propose a computational model of the recognition of scene categories that bypasses the segmentation and the processing of objects. With this point of view, they estimate the structure or "shape of a scene" using a few perceptual dimensions specifically dedicated to describe spatial properties of the scene. They show that holistic spatial scene properties, that they termed *Spatial Envelope properties*, may be reliably estimated using spectral and coarsely localized information. They claim that the scene representation characterized by the set of spatial envelope properties provides a meaningful description of the scene picture and its semantic category.

A remarkable characteristic of human visual system is that we are able to understand the meaning of a complex novel scene very quickly even when the image is blurred [35] or presented for only 20 msec [36]. In a study of Oliva and Torralba [37] the authors reference a work of Mary Potter [38,39] who demonstrated that during a rapid presentation of a stream of images, observers were able to identify the semantic category of each image as well as a few objects and their attributes. The amount of perceptual and semantic information that observers comprehend within a *glance* (about 200 msec) refers to the *gist* of the scene [40]. It is fundamental to understand what visual information is perceived during the course of this *glance*.

There has been numerous work and investigation in the domain of cognitive science focusing on rapidness and robustness of human scene categorization. There has been much work on scene understanding domain too. Research on the scene understanding domain has considered objects as atoms of recognition. On the other hand behavioral experiments on fast scene perception suggest an alternative view: that we do not need to perceive the objects in a scene to identify its semantic category. Humans can understand the semantic category of a real world scene from its spatial layout (a number of previous work about the spatial layout of a scene is referenced in [37]).

Figure 3.1 Spatial Layout

Figure 3.1 illustrates that the spatial layout ('spatial arrangement' or 'spatial relationships') of regions for scene and object recognition is very important. When looking at the image on the left, viewers describe the scene as a street with cars, buildings and the sky. The local information available in the image is insufficient for reliable object recognition; even so viewers are confident and highly consistent in their descriptions. Indeed, the blurred scene has the spatial layout of a street. When the image is shown in high resolution, new details reveal that the image has been manipulated and that the buildings are in fact pieces of furniture. Almost 30% of the image pixels correspond to an indoor scene. The misinterpretation of the low-resolution image is not a defect of the visual system. Instead, it illustrates the strength of spatial layout information in constraining the identity of the objects in normal conditions, which is especially evident in degraded conditions in which object identities cannot be inferred based on local information. This experiment demonstrates that global information of the image is very important for human perception and cognition. We use the global information for identifying a real world scene and the objects in it.

Based on the previous research of Navon [41] and the review of Kimchi [42] Oliva and Torralba point out that the processing of the global structure and the spatial

relationships between components precede the analysis of local details. The global precedence effect is particularly strong for images constituted of many element patterns [43] as it is the case of most real world scene pictures. Authors define a 'holistic cue' as 'one that is processed over the entire visual field and does not require attention to analyze local details'.

Renninger and Malik [44] conducted a research on scene perception. Their objective was to find out what sort of representation or information we are using to identify scenes so quickly. According to their literature survey, Friedman (1979) [45] proposed that the visual system might first recognize a ''diagnostic object'' that in turn triggers recognition of the scene. For example, a toaster would be diagnostic of a kitchen scene. Others argue that scenes may have distinctive holistic properties. For example, Biederman (1972) [46] found that subjects have more difficulty recognizing and locating objects in a jumbled scene than in a coherent one, even when the objects remain intact. Loftus, Nelson, and Kallman (1983) [47] studied the availability of holistic versus specific feature cues in picture recognition experiments. For brief presentations, subjects performed better when their response depended on the holistic cue. The arguments for a holistic property are consistent with the fact that we do not need to scan an image with our eyes or apply attention to particular objects in order to get the gist of the scene and most research supports this theory (Loftus et al., 1983; Metzger & Antes, 1983 [48] ; Schyns & Oliva, 1994 [35]). Renninger and Malik focus on the role of texture as a holistic cue.

One of the studies that focus on automatic scene categorization is the one of Vogel et al. They investigated on processing of local and global information in scene categorization. Their research focused on the processes underlying human scene categorization that would enable efficient computer vision systems. In a set of human experiments, categorization performance is tested when only local or only global image information is present. In the experiment focusing on local information, global configural information was eliminated by cutting the scenes into local image regions and randomly relocating, i.e. scrambling, those local regions (see Figure 3.2). In the experiment focusing on global information, local information was eliminated not only by low pass

filtering the images but also gray-scaling those to create stimuli that contain only global configural information (Figure 3.2).

Figure 3.2 a) intact b) scrambled c)blurred

Their 3 experiments are as follows: In the first experiments they present intact images to the participants, in the second and the third they present respectively scrambled and blurred images. Participants are required to check one of the five checkboxes labeled coasts, rivers/lakes, forests, plains, and mountains. Display time was 4 seconds after which subjects were forced to make a choice.

Table 3.1 shows the confusion matrix for categorization for intact images. The averaged over all subjects and all scene categories, the categorization rate was 89.7%

Table 3.1 Confusion matrix for categorization of intact images.

89.7%	coast	rivers	forest	plains	mountains
coast	90.4%	8.3%	0.3%	0.3%	0.6%
rivers	6%	82.9%	2.1%	0.4%	8.7%
forest	0.4%	1.6%	91.5%	4.7%	1.8%
1plains	0.4%	0%	0.8%	92.7%	6.1%
mountains	0.2%	2.9%	1.4%	5.0%	90.6%

Table 3.2 shows the confusion matrix of the categorization (see also Figure 3.2). The categorization performance is surprisingly good given that the important configural information has been eliminated.

Table 3.2 Confusion matrix for categorization of scrambled images.

72.7%	coast	rivers	forest	plains	mountains
coast	71.8%	14.2%	2.6%	3.5%	8.0%
rivers	18.8%	36.8%	16.3%	5.0%	23.1%
forest	0.9%	1.5%	91.3%	5.3%	1.1%
plains	0.8%	0.8%	2.8%	87.0%	8.7%
mountains	4.6%	2.7%	6.9%	12.3%	73.4%

Table 3.3 reveals that compared to table 3.2 there are fewer confusions between rivers/lakes and coasts, rivers/lakes and forests, and mountains and plains, but that there are now more confusions between coasts and mountains, plains and mountains, and plains and rivers/lakes.

Table 3.3 Confusion matrix for categorization of blurred images

71.6%	coast	rivers	forest	plains	mountains
coast	63.3%	14.0%	3.8%	5.6%	13.4%
rivers	8.7%	53.9%	8.7%	5.8%	22.9%
forest	0.9%	4.9%	86.4%	2.4%	5.4%
plains	4.0%	7.5%	3.8%	72.1%	12.6%
mountains	2.6%	5.2%	5.1%	6.2%	81.0%

Their results suggest that local and global information is integrated differently depending on the category. Categories with many different local semantic concepts

present in an image (such as mountains or rivers/lakes) require global context information for categorization. In contrast, categories such as forests, plains, or coasts with local semantic concepts that are discriminant without global configural information are categorized better using local information. Interestingly, the performance for intact scenes was higher than the performance in the scrambled and blurred conditions. This is consistent with the view that processing of local and global information are integrated resulting in higher categorization performance (see Figure 3.3)

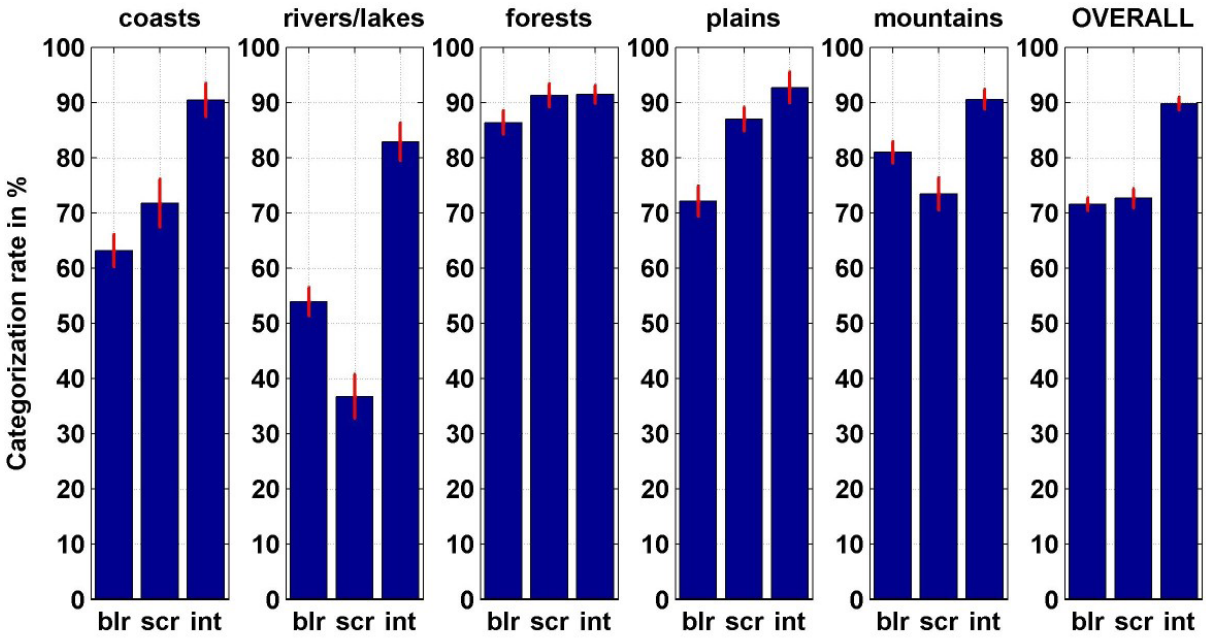

Figure 3.3 Comparison of categorization rates between blurred (blr), scrambled (scr), and intact (int) display condition (in%).

3.3 Intermediate Representation of Images

Efficient representation of images is a very important issue for various topics in image processing including scene understanding and categorization. In the previous section we have seen global and local representation of images. Another approach to image representation is namely intermediate representation which consists of building a dictionary for image categories. This dictionary (also called visual vocabulary) contains image patches (visual words) as words. An image is represented in terms of indexes of

this dictionary. Much of the previous work on image classification is based on this representation approach analogous to the *bag-of-words* model for text document retrieval. An illustration of this model is shown in figure 3.4. In the left side of the figure we see a face, in the right side; we see the parts of the face image.

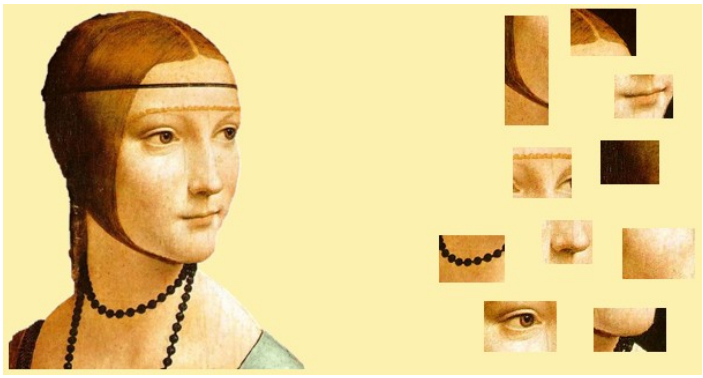

Figure 3.4 A face image and the parts of it

For the sake of clarity, here is an example of three images (figure 3.5) and the dictionary (figure 3.6) made of the parts of this these images:

Figure 3.5 Face, Bicycle, Violin

Figure 3.6 A dictionary containing parts of face, bicycle and violin.

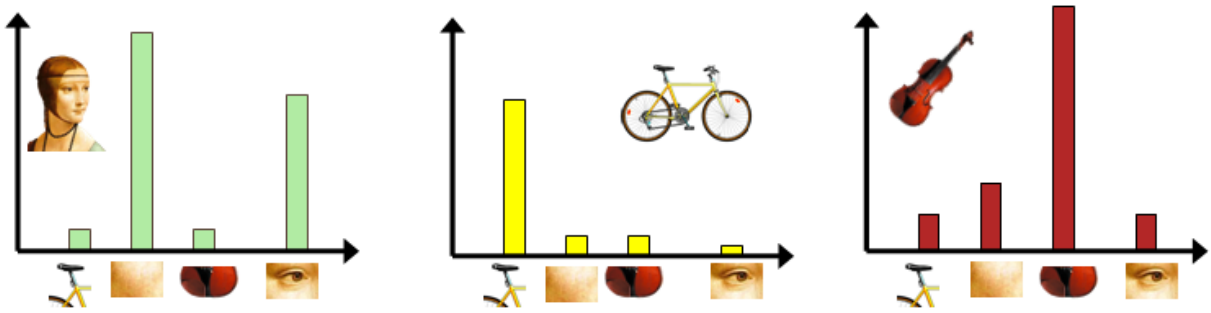

Figure 3.7 Histograms of the images

Figure 3.7 shows the histogram representation of the three images. Details of the method and the detail about obtaining image patches are presented in section 3.8.

In [49], Fei-Fei and Perrona have studied the literature on natural scene categorization; they are particularly interested in works where experiments showing that humans are capable of categorizing complex natural scenes very quickly. In their survey they cited works where categorization is performed using global cues (e.g. power spectrum, color histogram information [30,50]). In turn they represented images as a collection of patches each assigned membership to a large dictionary of codewords. Their image dataset contains the following categories of natural scenes: Highway, inside of cities, tall buildings, streets, suburb residence, forest, coast, mountain, open country, bedroom kitchen, living room and office. And they use 40 intermediate themes to represent images (e.g. sky, foliage, rock, etc.)

Vogel and Schiele [7] also used an intermediate representation obtained from human observers. They work on coasts, rivers/lakes, forests, plains, mountains and sky/clouds categories and they use sky, water, grass, trunks, foliage, field, rocks, flowers and sand as semantic concepts to represent an image. Each image is represented by a concept occurrence vector which tabulates the frequency of occurrence of each local semantic concept. In [7], human subjects are asked to classify 59, 582 local patches from the training images into one of 9 different "semantic concepts" (e.g. water, foliage, sky, etc.). In [51] Bosh et al. applied to a bag of visual words representation for images. They used the same datasets in [7,28,49]. They term the intermediate representation as 'topic'. In their case, they discover between 22 and 30 topics for 8 categories. These

topics can vary depending the color features (where topics can distinguish objects with deferent colors like *light sky*, *blue sky*, *orange sky*, *orange foliage*, *green foliage* etc...) or only grey SIFT features (objects like *trees* and *foliage*, *sea*, *buildings* etc...). These works clearly point to the usefulness of these intermediate representations.

3.4 Color Features

Color is usually the first descriptor used to represent an image. A widely used technique for color descriptor is the color histogram. Histograms are easily and rapidly calculated and they are robust to rotation and translation. However the use of histograms for image categorization, indexing and retrieval cause some problems [52]. First problem is the size of the histograms that makes the use of it difficult. Secondly histograms do not contain spatial information about the positions of the color. Third problem is that they are very sensible in little differentiation of the luminosity which causes problems comparing images of the same scene taken in different condition (e.g day and night). There has been two approaches to solve these problems one of these approaches is to add spatial information to histograms. Stricker et al. [53] divided the image in five blocks fixed superposed and they extracted the three first moments of inertia for each block to create descriptor vectors. Pass and Zabih [54] added the spatial coherence in the histograms introducing a color feature called color coherence vector. Huang et al [55] proposed the correlogramme and autocorrelogramme. Paschos and Radev introduced a feature called chromaticity moment [56]. Shih and Chen described color moments as color feature [57].

The second approach search other color spaces based on the color perception of human. The RGB color space is widely used in every computer vision system because of its ease of use but it is not the best adapted to human visual system. In fact the three components R, G and B are very dependent between each other. A slide change of luminosity modifies the three components given that the objects in that scene keep their original color, but they are simply lightened. Smeulders et al. presented other color spaces in their experimentation. Park et al. [58] proposed the CIE LUV color space. Gong et al. [59] calculated histograms on HSV (Hue Saturation Value) color space.

3.4.1 Color Histograms

In computer graphics a color histogram is representation of the distribution of colors in an image. Computationally, the color histogram is formed by discrediting the colors within an image and counting the number of pixels of each color. Let a color space $x_1 x_2 x_3$ discredited in n levels for each x_i i=1 to 3. Then there are n distinct colors for each x_i. A color $\vec{c}$ is a vector defined as: $\vec{c}$ = x. $\vec{x}_1$ + y. $\vec{x}_2$ + z. $\vec{x}_3$

Let *I* an image, the histogram H (*I*)of this image is defined as:

$$H=<hx_1^1,\ hx_1^2,\ldots,hx_1^n,hx_2^1,hx_2^2,\ldots,hx_2^n,hx_3^1,hx_3^2\ldots,hx_3^n> \quad (3.1)$$

Where each hx_j^i contains the number of pixels of color j of level I in the image.

3.4.2 Color Correlogram

The highlights of this feature are:

(i) it includes the spatial correlation of colors,

(ii) it can be used to describe the global distribution of local spatial correlation of colors;

(iii) it is easy to compute, and

(iv) the size of the feature is fairly small.

Informally, a color correlogram of an image is a table indexed by color pairs, where the k-th entry for <i,j> specifies the probability of finding a pixel of color j at a distance k from a pixel of color i in the image. Such an image feature turns out to be robust in tolerating large changes in appearance of the same scene caused by changes in viewing positions, changes in the background scene, partial occlusions, camera zoom that causes radical changes in shape, etc.

A color correlogram expresses how the spatial correlation of pairs of colors changes with distance. A color histogram captures only the color distribution in an image and

does not include any spatial correlation information. Let *I* be an nxn image. The colors in *I* are quantized into m colors ($c_1,\ldots, c_m$). For a pixel $p = (x, y) \in I$, let *I*(p) denote its color. Let I_c = { p | I(p)=c } Thus, the notation $p \in I_c$ is synonymous with $p \in I$, I(p)=c. For convenience, we use the L∞ norm to measure the distance between pixels, p1 = (x1,y1), p2 = (x2,y2), we define |p1-p2|=max{ |x1-x2|, |y1-y2|}. We denote the set {1,2,…,n} by [n]. The *histogram* h of I is defined for $i \in [m]$ by:

$$h_{c_i}(I) = n^2 \cdot \Pr_{p \in I}\left[p \in I_{c_i}\right] \tag{3.2}$$

For any pixel in the image, $h_{c_i}(I) / n^2$ gives the probability that the color of the pixel is c_i. Let a distance $d \in [n]$ be fixed *a priori.* Then, the *correlogram* of *I* is defined for $k \in [d]$, $j \in [m]$ as:

$$\gamma^{k}_{c_i,c_j}(I) = \Pr_{p_1 \in I_c, p_2 \in I}\left[p_2 \in I_{c_j} \setminus |p_1 - p_2| = k\right] \tag{3.3}$$

Given any pixel of color c_i in the image, $\gamma^{(k)}{c_i, c_j}$ gives the probability that a pixel at distance k away from the given pixel is of color c_j. Note that the size of the correlogram is O(m²d). The *autocorrelogram* of *I* captures spatial correlation between identical colors only and is defined by:

$$\alpha_c^{(k)}(I) = \gamma_{c,c}^{(k)}(I) \tag{3.4}$$

3.4.3 Color Coherence Vector

Intuitively, a color's *coherence* is defined as the degree to which pixels of that color are members of large similarly-colored regions. These significant regions are referred as *coherent regions*, and it has been observed that they are of significant importance in characterizing images. The coherence measure classifies pixels as either coherent or incoherent. Coherent pixels are a part of some sizable contiguous region, while incoherent pixels are not. A *color coherence vector* represents this classification for each color in the image. CCV's prevent coherent pixels in one image from matching incoherent pixels in another. This allows fine distinctions that cannot be made with color histograms.

3.4.3.1 Computation of CCV

First step is blurring the image in order to eliminate small variations between neighboring pixels. Second step is discrediting the color space, such that there are only n distinct colors in the image. The next step is to classify the pixels within a given color bucket as either coherent or incoherent. A coherent pixel is part of a large group of pixels of the same color, while an incoherent pixel is not. We determine the pixel groups by computing connected components. A connected component C is a maximal set of pixels such that for any two pixels p, $p' \in C$, there is a path in C between p and p'. (Formally, a path in C is a sequence of pixels $p=p_1,\ldots,p_2$, $p_n = p'$ such that each pixel p_i is in C and any two sequential pixels p_i, p_{i+1} are adjacent to each other. We consider two pixels to be adjacent if one pixel is among the eight closest neighbors of the other. In other words, we include diagonal neighbors.

When connected components are computed, each pixel will belong to exactly one connected component. We classify pixels as either coherent or incoherent depending on the size in pixels of its connected component. A pixel is coherent if the size of its connected component exceeds a fixed value τ, otherwise, the pixel is incoherent. For a given discretized color, some of the pixels with that color will be coherent and some will be incoherent. Let us call the number of coherent pixels of the j^{th} discretized color α_j and the number of incoherent pixels β_j. Clearly, the total number of pixels with that color is $\alpha_j+\beta_j$, and so a color histogram would summarize an image as:

$$< \alpha_1 + \beta_1;...;\alpha_n + \beta_n > \tag{3.5}$$

Instead, for each color we compute the pair (α_j,β_j) which we will call the *coherence pair* for the j^{th} color. The *color coherence vector* for the image consists of

$$< (\alpha_1, \beta_1),...,(\alpha_n, \beta_n) > \tag{3.6}$$

This is a vector of coherence pairs, one for each discretized color. An example for calculation of CCV is given below:

Let $\tau = 4$ and assume that we work with an image in which all 3 color components have the same value at every pixel (in the RGB color space this would represent a grayscale image). This allows us to represent a pixel's color with a single number (i.e., the pixel with R/G/B values 12/12/12 will be written as 12). Suppose that after we slightly blur the input image, the resulting intensities are as follows:

22	10	21	22	15	16
24	21	13	20	14	17
23	17	38	23	17	16
25	25	22	14	15	21
27	22	12	11	21	20
24	21	10	12	22	23

Let us discretize the colorspace so that bucket 1 contains intensities 10 through 19, bucket 2 contains 20 through 29, etc. Then after discretization we obtain:

2	1	2	2	1	1
2	2	1	2	1	1
2	1	3	2	1	1
2	2	2	1	1	2
2	2	1	1	2	2
2	2	1	1	2	2

The next step is to compute the connected components. Individual components will be labeled with letters (A, B,…) and we will need to keep a table which maintains the discretized color associated with each label, along with the number of pixels with that label. Of course, the same discretized color can be associated with different labels if multiple contiguous regions of the same color exist. The image may then become:

$$\begin{matrix} B & C & B & B & A & A \\ B & B & C & B & A & A \\ B & C & D & B & A & A \\ B & B & B & A & A & E \\ B & B & A & A & E & E \\ B & B & A & A & E & E \end{matrix}$$

And the connected components table will be:

Table 3.4 Connected components

Label	A	B	C	D	E
Color	1	2	1	3	1
Size	12	15	3	1	5

The components A, B, and E have more than pixels, and the components C and D less than τ pixels. Therefore the pixels in A, B and E are classified as coherent, while the pixels in C and D are classified as incoherent. The CCV for this image will be:

Table 3.5 CCV example

Color	1	2	3
α	17	15	0
β	3	0	1

A given color bucket may thus contain only coherent pixels (as does 2), only incoherent pixels (as does 3), or a mixture of coherent and incoherent pixels (as does 1). If we assume there are only 3 possible discretized colors, the CCV can also be written:

<(17; 3) ; (15; 0) ; (0; 1)> .

3.4.4 Color Moments

The color distributions of the R**,** G and B components of an image can be represented by its color moments. The first color moment of the ith color component (i=1, 2, 3) is defined by:

$$M_i^1 = \frac{1}{N}\sum_{j=1}^{N} p_{i,j} \tag{3.7}$$

Where $p_{i,j}$ is the color value of the ith color component of the jth image pixel and *N* is the total number of pixels in the image. The hth moment, *h* = 2, ***3,*** . . . , of the ith color component is then defined as:

$$M_i^h = \left(\frac{1}{N}\sum_{j=1}^{N}\left(p_{i,j} - M_i^1\right)^h\right)^{1/h} \tag{3.8}$$

Take the first *H* moments of each color component in an image s to form a feature vector, CT, which is defined as:

$$CT=[ct_1, ct_2, \ldots ct_Z]$$
$$=[\alpha_1 M_1^1, \alpha_1 M_1^2, \ldots, \alpha_1 M_1^H, \alpha_2 M_2^1, \alpha_2 M_2^2, \ldots, \alpha_2 M_2^H, \alpha_3 M_3^1, \alpha_3 M_3^2, \ldots, \alpha_3 M_3^H] \tag{3.9}$$

Where *Z= 3xH* and α_1, α_2**,** α_3, are the weights for the R,G,B components. Based on the above definition, an image is first divided into *X* non-overlapping blocks. For each

block ***a,*** its h[th] color moment of the i[th] color component is defined by $M_{a,i}^{h}$ Then, the feature vector CB_a of block *a* is represented as:

$$CB_a = [\, cb_{a,1}, cb_{a,2}, \ldots, cb_{a,Z} \,]$$
$$= [\, \alpha_1 M_{a,1}^{1}, \alpha_1 M_{a,1}^{2}, \ldots, \alpha_1 M_{a,1}^{H}, \alpha_2 M_{a,2}^{1}, \alpha_2 M_{a,2}^{2}, \ldots, \alpha_2 M_{a,2}^{H}, \alpha_3 M_{a,3}^{1}, \alpha_3 M_{a,3}^{2}, \ldots, \alpha_3 M_{a,3}^{H} \,] \quad (3.10)$$

From the above definition we can obtain *X* feature vectors for an image. Color moments have been proved to be efficient and effective in representing color distribution of images in a very compact way but this compactness can lower the discrimination power of these features [4].

3.4.5 Chromaticity Moments

The concept of the xy chromaticity diagram is defined within the xyY color space, which is an extension of the CIE XY Z space. From each image pixel a pair of (x,y) chromaticity values is derived, thus, leading to a unique set of chromaticities for a given image. This chromaticity set is characterized by two attributes:

(a) its two dimensional shape on the x - y space,

(b) its three dimensional distribution (i.e., histogram) over the x - y space (in general, the same chromaticity values will be produced by more than one pixels in the same image).

Given that x,y $\in$ [0,1], one will need to quantize x and y to appropriately chosen levels, X_S and Y_S, respectively. Thus, the chromaticity diagram is a two-dimensional representation of an image. Each pixel (i.e., each (R,G,B) triplet) produces a pair of (x,y) chromaticities. Therefore an image yields a set of distinct (x,y) pairs, its chromaticity set. Accordingly, for an image I of dimensions L_x,L_y we define the trace of its chromaticity set as:

$$T(x,y)=\begin{cases}1 & if\ \exists(i,j): I(i,j)\ produces(x,y) \\ 0 & otherwise\end{cases} \qquad 0 \le i \le L_x - 1,\ 0 \le j \le L_y \qquad (3.11)$$

In addition, more than one pixels may produce the same (x,y) pair. Thus, the corresponding two-dimensional distribution (i.e., histogram) is defined as follows:

D(x,y) = #pixels yielding (x,y).

Each of these two functions T and D can be characterized, within approximation, by a set of moments, defined, respectively, as follows:

$$M_T(m,l)=\sum_{x=0}^{X_s-1}\sum_{y=0}^{Y_s-1} x^m y^l T(x,y) \qquad (3.12)$$

$$M_D(m,l)=\sum_{x=0}^{X_s-1}\sum_{y=0}^{Y_s-1} x^m y^l D(x,y) \qquad (3.13)$$

Where m = 0, 1, 2, . . . , l = 0, 1, 2, . . . , and X_s, Y_s are the dimensions of the xy space. M_T and M_D form the set of *chromaticity moments* of image *I*.

3.5 Texture Features

The texture is a visual feature studied the last two decades. Many techniques have been developed in order to analyze the texture. One of these methods is the very well known Haralick co-occurrence matrix. Four measures of these matrices are extracted and used. These measures are energy, entropy, contrast and homogeneity. There exist other

methods for analyzing texture based on Gabor filters. After applying the Gabor transformation to an image, a texture region is characterized with the mean and the variance of the coefficients of the transform. A feature vector is build using these features as components.

3.5.1 Co-Occurrence Matrix

A co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset. Mathematically, a co-occurrence matrix C is defined over a n x m image I, parameterized by an offset (Δx,Δy), as:

$$C(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & \textit{if} \quad I(p,q) = i \ \textit{and} \ \ I(p+\Delta x, q+\Delta y) = j \\ 0, & \textit{otherwise}\end{cases} \tag{3.14}$$

The elements of this matrix, *p(i,j)*, represent the relative frequency by which two pixels with grey levels "i" and "j", that are at a distance "d" in a given direction, are in the image or neighborhood. It is a symmetrical matrix, and its elements are expressed by:

$$p(i,j) = \frac{P(i,j)}{\sum_{i=0}^{N_g-1}\sum_{j=\grave{a}}^{N_g-1} P(i,j)} \tag{3.15}$$

Where *Ng* represents the total number of grey levels. Using this matrix, Haralick (1973) proposed several statistical features representing texture properties, like *contrast*, *uniformity*, *mean*, *variance*, *inertia moments*, etc. Some of those features were calculated, selected and used in this study.

3.5.2 Energy

Energy is a measure of textural uniformity in an image. Mathematically, it is given as:

$$Energy = \sum_i \sum_j I_{i,j}^2 \tag{3.16}$$

Where I is the image

3.5.3 Entropy

Entropy is a measure of disorder or complexity of the image.

$$Entropy = \sum_i \sum_j I_{i,j} \log I_{i,j} \tag{3.17}$$

Where I is the image

3.5.4 Homogeneity

Homogeneity is a measure of the overall smoothness of an image.

$$Homogeneit\ y = \sum_i \sum_j \frac{1}{1+(i-j)^2} I_{i,j} \tag{3.18}$$

Where I is the image

3.5.5 Wavelet Transform

The use of wavelets has developed in many fields for analyzing, synthesizing, de-noising, and compressing signals and images. The use of wavelet transform as a multiscale analysis for texture description was first suggested by Mallat [60]. Recent developments in the wavelet transform provide good analytical tool for texture analysis and can achieve a high accuracy rate. The discrete wavelet transform (DWT) is a simple and intuitive method to discriminate similar images. The wavelet transform provides a robust methodology for texture analysis in different scales. The wavelet transform allows for the decomposition of a signal using a series of elemental functions

called *wavelets* and *scaling*, which are created by scalings and translations of a base function, known as the *mother wavelet*:

$$\Psi_{s,u}(x) = \frac{1}{\sqrt{s}} \Psi\left(\frac{x-u}{s} \right) \tag{3.19}$$

where "*s*" governs the scaling and "*u*" the translation. The wavelet decomposition of a function is obtained by applying each of the elemental functions or wavelets to the original function:

$$Wf\ (s,u) = \int f(x) \frac{1}{\sqrt{s}} \Psi^* \left(\frac{x-u}{s} \right) dx \tag{3.20}$$

In practice, wavelets are applied as high-pass filters, while scalings are equal to low-pass filters. As a result of this, the wavelet transform decomposes the original image into a series of images with different scales, called trends and fluctuations. The former are averaged versions of the original image, and the latter contain the high frequencies at different scales or levels.

3.5.5.1 Discrete Wavelet Transform

The discrete wavelet transformation (DWT) decomposes an original signal f(x) with a family of basis functions $\psi_{m,n}(x)$: which are dilatations and translations of a single prototype wavelet function known as $\psi(x)$:

$$f(x) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} c_{m,n} \Psi_{m,n}(x) \tag{3.21}$$

$C_{m,n}$ constitutes the DWT coefficients where m and n are integers and referred as the dilation and translation parameters. An efficient way to implement this scheme using filters was developed by Mallat [60]. The 2D DWT is computed by a pyramid transform scheme using filter banks. The filter banks are composed of a low pass and a high pass filter and each filter bank is then sampled down at a half rate of the previous frequency. The input image is convolved by a high pass filter and a low pass filter in horizontal direction (rows). After this step another convolution in vertical direction (columns) is performed with a high and a low pass filter. Thus the original image is transformed into four sub images after each decomposition step. A three level decomposition results in 10 sub images, where t^h approximation image is the input image for the next level.

3.5.5.2 Texture Feature Extraction Using Gabor Function

A two dimensional Gabor function g(x, y) and its Fourier transform G(u, v) can be written as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi Wx\right] \tag{3.22}$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \tag{3.23}$$

Where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$

Gabor functions form a complete but non orthogonal basis set. Expanding a signal using this basis provides a localized frequency description. A class of self-similar functions, referred to as Gabor wavelets in the following discussion, is now considered. Let g(x, y) be the mother Gabor wavelet, then this self-similar filter dictionary can be obtained by appropriate dilations and rotations of g(x, y) through the generating function:

$$g_{mn}(x,y) = a^{-m}G(x',y'), \quad a>1, \tag{3.24}$$

$$x' = a^{-m}(x\cos\theta + y\sin\theta), \text{ and } y' = a^{-m}(-x\sin\theta + y\cos\theta), \tag{3.25}$$

Where $\theta = n\pi / K$ and K is the total number of orientations. The scale factor is a^{-m} meant to ensure that the energy is independent of m. Given an image I(x, y), its Gabor wavelet transform is then defined to be:

$$W_{mn}(x,y) = \int I(x_1,y_1)g_{mn}^{*}(x-x_1,y-y_1)dx_1dy_1, \tag{3.26}$$

where * indicates the complex conjugate. It is assumed that the local texture regions are spatially homogeneous, and the mean and the standard deviation of the magnitude of the transform coefficients are used to represent the region for classification and retrieval purposes:

$$\mu_{mn} = \iint |W_{mn}(xy)|dxdy \tag{3.27}$$

$$\sigma_{mn} = \sqrt{\iint \left(|W_{mn}(x,y)| - \mu_{mn}\right)^2 dxdy} \tag{3.28}$$

A feature vector is now constructed using μ_{mn} and σ_{mn} as feature components. If we use four scales S = 4 and six orientations K = 6, the resulting feature vector is as follows:

$$f = [\mu_{00}\sigma_{00}\mu_{01}...\mu_{35}\sigma_{35}] \tag{3.29}$$

3.6 Edge Features

Edge features are very important for image representation. Two of these different edge features are edge histograms and edge direction coherence vectors. Vailaya et. al point out that Man-made objects in the city scenes usually have strong vertical and horizontal edges, whereas non-city scenes tend to have edges randomly distributed in various directions. A feature based on the distribution of edge directions can discriminate between the two categories of images.

3.6.1 Edge Histogram Description (EHD)

The edge histogram descriptor (EHD) is defined in the texture part of the MPEG-7 standard [61]. The distribution of edges is not only a good texture signature; it is also useful for image-to- image matching in the absence of any homogeneous texture. A given image is first divided into 16 sub-images 4 x 4; and local edge histograms are computed for each sub-image. To compute the edge histogram, each of the 16 sub-images is further subdivided into image blocks. Note that, regardless of the image size, we divide the sub-image into a fixed number of image-blocks. That is, the size of the image-block is proportional to the size of original image to deal with the images with different resolutions. The size of each image block is proportional to the size of the original image and is assumed to be a multiple of two. The number of image blocks, independent of the original image size, is constant (desired_num_of_blocks) and the block size is figured as follows:

$$x = \sqrt{\frac{image\_width * image\_height}{desired\_num\_blocks}} \tag{3.30}$$

$$block\_size = \left\lfloor \frac{x}{2} \right\rfloor * 2 \tag{3.31}$$

Where image_width and image_height represent the horizontal and vertical size of the image, respectively. Each image block is then partitioned into four 2 x 2 blocks of pixels, and the pixel intensities for these four divisions are computed by averaging the luminance values of the existing pixels. In the case of edge images, the luminance takes only the value of one or zero. Edges are grouped into five classes: vertical, horizontal, 45 diagonal, 135 diagonal and isotropic (non-directional) based on directional edge strengths. These directions are determined for each image block using five corresponding 2 x 2 filter masks corresponding to 2 x 2 sub-divisions of the image blocks. If the maximum directional strength is greater than a threshold value Th_{edge} then the underlying block is designated to belong to the corresponding edge class. The default value of Th_{edge} for grey-scale images is 11 and for binary edge images we set it to zero. The histogram for each sub-image represents the frequency of occurrence of the five classes of edges in the corresponding sub-image. As there are 16 sub-images and each has a five-bin histogram, a total of 16 x 5 = 80 bins in the histogram is achieved. For normalization, the number of edge occurrences for each bin is divided by the total number of image blocks in the sub-image. To minimize the overall number of bits, the normalized bins are nonlinearly quantized and fixed-length coded with 3 bits per bin, resulting in a descriptor of size 240 bits.

3.6.2 Histogram of Edge Direction

Histograms of edge directions are translation invariant and capture the general shape information in the images. Because the feature is local, it is robust to partial occlusion and local disturbance in the image. The edges and their directions are calculated using "Canny" [62] edge operator and only the strong edges are retained, i.e., edges that lie in a segment of a similar direction. Since the calculated edge directions are not totally accurate a quantization is made to the edge directions into a few directions (e.g. four directions: horizontal, vertical, and the two diagonals (top right to bottom left and top left to bottom right). In this case the histogram representation for an image or a region of the image would have only four bins.).

The problem with presenting an image with only one histogram of edge directions is that it preserves only global information about the directions of the edges in the image.

In this case it is possible to have two images with totally different edge patterns and have the same histogram of edge directions. To overcome this problem we decompose each image into *m x* n rectangular regions *(N* regions) and represent each region by a histogram. Then, each histogram is normalized with respect to the total number of edges in the region. The normalization is important because an image with the same content but at a different scale will produce a different histogram but a similar normalized histogram [63].

Vailaya et. al. [3] used edge direction histogram as a texture measure they used edge direction histogram quantizing at 5 degree intervals from 0 to 360. Thus 72 bins are used to represent the edge directions. A 73rd bin is also added which represents the non-edge pixels in the image. The histograms are normalized as follows:

$$H(i) = \frac{H(i)}{n_e} \tag{3.32}$$

Where H(i) is the edge direction histogram and n_e is the total number of edge points detected in the image.

Brandt et al. calculate the edge direction histogram as follows: At first, the color image is transformed to the HSI space from which the hue channel is neglected. The other two channels are convolved with the eight Sobel operators. The resulting gradient images are next thresholded to binary images by a proper threshold value for each channel. The threshold values are manually fixed to certain levels which are the same for all images. The thresholded intensity and saturation gradient images are combined by the logical OR operation. The threshold value for the intensity gradient image was manually set to 15% of the maximum gradient value and for the saturation image to 35%. In the OR operation, the direction of the larger gradient value is chosen. Finally the 8-dimensional edge histograms are calculated by counting the edge pixels in each direction. Still, it is necessary to normalize the histograms somehow. They show by an experiment that it is

better to normalize the histograms by the number of pixels in each image rather than by number of edge pixels as was done in [58]. They studied also the effect of smoothing proposed in [58]. The smoothing makes the histograms more robust to rotation. It is performed as follows:

$$I_s[i] = \frac{\sum_{j=i-k}^{i+k} I[j]}{2k+1} \tag{3.33}$$

Where I_s is the smoothed histogram, I is the original normalized histogram and the parameter k determines the degree of smoothing.

Jain and Vailaya propose an edge direction histogram for image retrieval [59] and employ it for trademark registration process. Shih and Chen [60] used histograms of edge direction to describe the shapes of representative objects in different trademarks Yoo et al. [61] apply the same histogram for shape representation in a new proposed content-based retrieval system.

3.6.3 Edge Direction Coherence Vector

An edge direction coherence vector stores the number of coherent versus incoherent edge pixels with the same directions (within a quantization of 5 degree). A threshold (0,1% of image size) on the size of every connected component of edges in a given direction is used to decide whether the region is coherent or not. This feature is thus geared towards discriminating structured edges from randomly distributed edges when the edge direction histograms are similar. Hauptmann et al used this feature to distinguished structured edges (like edges of buildings) from arbitrary edge distributions [65]. Vailaya et al. show that edge direction coherence vector perform better then edge direction histogram, color histogram, and color coherence vector for classifying city vs. landscape images

3.7 Spectral Features

3.7.1 Power Spectrum of an Image

The power spectrum of an image is computed by taking the squared magnitude of its Fourier Transform:

$$\Gamma\left(f_x, f_y\right) = \left|FT\left\{i\left(x, y\right)\right\}\right|^2 \tag{3.34}$$

where *I(x, y)* is the intensity distribution of the image along the spatial variables x and y. FT is the Fourier Transform, f_x and f_y are the spatial frequencies. Power spectrum, $\Gamma(f_x,f_y)$, encodes the energy density for each spatial frequency and orientations over the whole image.

3.7.2 Gist of an Image

Gist is a global feature proposed by Oliva and Torralba [62]. It is calculated using power spectrum of the image and a series of Gabor filters as follows:

$$g_n = \iint \Gamma\left(f_x, f_y\right) G_n\left(f_x, f_y\right)^2 df_x df_y \tag{3.35}$$

An image is represented by a feature vector x = gn, gn being the output energies of a set of Gabor filters.

3.8 Scale-Invariant Feature Transform (SIFT)

David G. Lowe presented a method for image feature generation called the Scale Invariant Feature Transform (SIFT). This approach transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. The steps of this method are as follows:

Scale-space extrema detection is the first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation. Keypoint localization each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability. One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

3.9 Bag of Words

The Bag-Of-Words (BoW) is a model originated of Natural Language Processing. According its original definition, it is a model used representing documents. It ignores the word orders. For example, the series of words "a beautiful day" and "day a beautiful" are exactly identical for this model. The BoW model produces a dictionary based modeling, and each document is similar to a "bag" (therefore the order of words is irrelevant), that contains words from the dictionary.

An example of BoW

Given these two text documents:

- “Battlestar Galactica has an FTL drive”
- “FTL drive of Battlestar Pegasus is broken”

A dictionary is constructed as follows:

- dictionary={1:" Battlestar ", 2:" Galactica ", 3:" has ", 4:" an ", 5:" FTL ", 6:"drive", 7:" of ", 8:" Pegasus ", 9:" is ", 10:" broken "},

This dictionary contains 10 words.

Using the indexes of the dictionary, The documents are represented by a 10-entry vector:

- [*1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 7:0, 8:0, 9:0, 10:0*]

- [*1:1, 2:0, 3:0, 4:0, 5:1, 6:1, 7:1, 8:1, 9:1, 10:1*]

where each indexed component of the vectors refers to count of the corresponding word in the dictionary. As it can be seen, the vector representation does not preserve the order. An application of this representation using Latent Dirichlet Allocation is presented in [70]

In Computer Vision, researchers use the BoW model for image representation. For instance, an image can be considered as a document, and features extracted from this image are considered as the words that it contains.
In order to represent an image using BoW model, the image is considered as a document. But the question is what is represented as words? The answer of this question is not trivial. It is laborious to define 'words' of an image. It includes feature detection and feature description steps. According to Fei-Fei et. al. a definition of the BoW model can be the "histogram representation based on independent features" [71].

Feature detection for the BoW model consists of extracting several local patches, which are considered as candidates for basic elements, "words". The feature detection step is in fact a selection of key points or regions in an image. In [49,72] this method is used as it is described. In [74] every pixel is considered as key point. A dense regular grid (10 to 30 pixels) in [86 and 90] shown in figure 3.16 (b) , randomly sampled points, segmentation based patches in [49], and sparse sets of interest points or regions, including Lowe's difference-of-Gaussians (DoG) peaks in [69] shown in Figure 3.8 (c).

Figure 3.8 (a)original image (b) regular grid (c) interest points

As a result of feature detection step, an image is abstracted by a set of locations (see Figure 3.8 (a) and (b). Then feature description methods take over for representation of the patches as numerical feature vectors. In other words, once a set of locations is obtained, local descriptors are extracted. A good descriptor should have the ability to handle intensity, rotation, scale and affine variations.

One of the most widely used local descriptors is SIFT [75], which is essentially a histogram of intensity gradient orientations, weighted by their magnitude and a Gaussian window. It is computed at different image scales, and the predominant gradient orientation is subtracted, to make it scale and rotation invariant. Descriptors are also often computed by passing an image through filter banks, typically comprising of Gaussians, Gaussian derivatives, Laplacians, and wavelets [76]. It is demonstrated that SIFT descriptors seem are more robust than other descriptors, and dense sampling grids outperform other point detectors [77]. After this step, each image is a collection of vectors of the same dimension (128 for SIFT), where the order of different vectors is of no importance. Figure 3.9 illustrates SIFT descriptors obtained from an image.

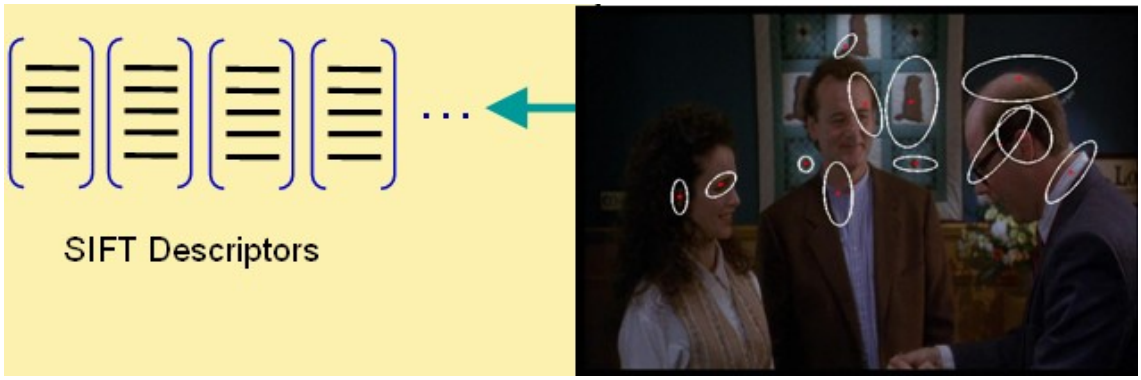

Figure 3.9 SIFT descriptor vectors

The next and final step is to convert vector represented patches (descriptors) to codewords. The collection of descriptors is vector-quantized into a dictionary of codewords (Shown in Figure 3.10).

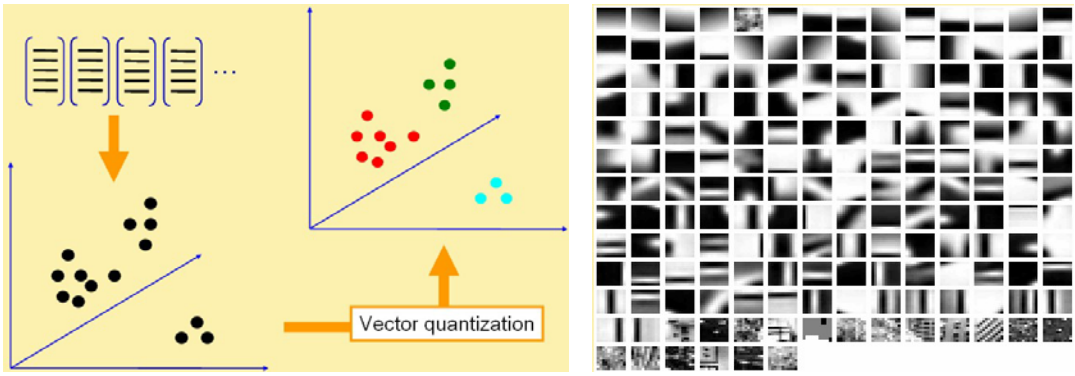

Figure 3.10 Vector quantization of descriptors and the dictionary

A codeword can be thought of a representative of several similar patches. One simple method is using K-means algorithm over all the vectors. Codewords are then defined as the centers of the learned clusters. The number of the clusters is the codebook size. The optimal dictionary size and codewords are learned by pairwise merging from an initially large dictionary. An image is the represented using a histogram of the codewords it contains (see Figure 3.11).

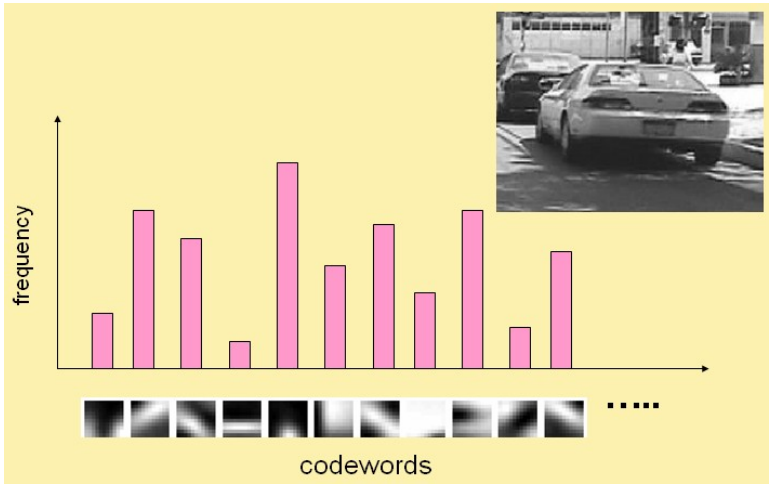

Figure 3.11 Codeword histogram representation of an image

A recapitulation of the BoW method:

(i) A set of key points/regions (patches) is selected,
(ii) Patches are represented using local descriptors,
(iii) Descriptors are vector quantized into a fixed-size codebook,
(iv) The image is represented as a histogram of the codewords it contains.

3.10 Conclusion

Three different approaches for image representation namely global, local and intermediate were studied and discussed. We have seen that all these three ways of representing images were equally important and that human perception can only be modeled by investigating in all these approaches, not neglecting one of them

4 IMAGE COMPRESSION

4.1 Introduction

Image Compression became more and more important in the last decade with the development of Internet and digital cameras becoming widespread. Uncompressed multimedia data like image audio and video requires large and larger storage capacity and transmission bandwidth every day. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. This recent growth of multimedia-based web applications has not only sustained the need for more efficient ways to compress images but have made compression central to storage and communication technology.

There is a certain number of features that an image compression system has to address, these features are: Lossless and lossy compression, embedded lossy to lossless coding, progressive transmission by pixel accuracy and by resolution, robustness to the presence of bit-errors and region-of-interest coding. An image compression system has to provide these features without degrading the image quality. To address these needs and requirements in the specific area of still image compression, many efficient techniques with considerably different features have recently been developed [78, 79]

Lossy CS, which aim at obtaining the best possible fidelity for a given bit-rate (or minimizing the bit-rate to achieve a given fidelity measure). Lossless CS aim at minimizing the bit rate of the compressed output without any distortion of the image. The decompressed bit-stream is identical to original bit-stream

4.2 Image Compression Standards

4.2.1 JPEG and JPEG2000

Compression is one of the technologies that enable the multimedia revolution to occur. However for technology to be effective there has to be some degree of standardization so that the equipment designed by different vendors can interoperate. International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) have been working together to establish a joint international standard for the compression of grayscale and color still images. This effort has been known as JPEG, the Joint Photographic Experts Group the "joint" in JPEG refers to the collaboration between ITU and ISO). Officially, JPEG corresponds to the ISO/IEC international standard 10928-1, digital compression and coding of continuous-tone (multilevel) still images or to the ITU-T Recommendation T.81. The text in both these ISO and ITU-T documents is identical. The process was such that, after evaluating a number of coding schemes, the JPEG members selected a DCT1-based method in 1988. From 1988 to 1990, the JPEG group continued its work by simulating, testing and documenting the algorithm. JPEG became a Draft International Standard (DIS) in 1991 and an International Standard (IS) in 1992. With the continual expansion of multimedia and Internet applications, the needs and requirements of the technologies used, grew and evolved. In March 1997 a new call for contributions were launched for the development of a new standard for the compression of still images, the JPEG2000. This project was intended to create a new image coding system for different types of still images (bi-level, gray-level, color, multi-component) [80] JPEG 2000 has been published as an ISO standard in 2000.

4.2.2 Other Standards

4.2.2.1 Graphics Interchange Format (GIF)

GIF images are compressed using the Lempel-Ziv-Welch (LZW) [81] lossless data compression technique to reduce the file size without degrading the visual quality. This compression technique was patented in 1985. Though the relevant patents have all since expired, the controversy over the licensing agreement between the patent holder,

Unisys, and CompuServe in 1994 led to the development of the Portable Network Graphics (PNG) standard.

4.2.2.2 Portable Network Graphics (PNG)

Portable Network Graphics (PNG) [82] is a W3C recommendation for coding of still images which has been elaborated as a patent free replacement for GIF, while incorporating more features than this last one. It is based on a predictive scheme and entropy coding. The prediction is done on the three nearest causal neighbors and there are five predictors that can be selected on a line-by-line basis. The entropy coding uses the Deflate algorithm of the popular Zip file compression utility, which is based on LZ77 coupled with Huffman coding. PNG is capable of lossless compression only and supports gray scale, palletized color and true color, an optional alpha plane, interlacing and other features. [83]

4.2.2.3 JBIG

JBIG is a lossless image compression standard from the Joint Bi-level Image Experts Group, standardized as ISO/IEC standard 11544 and as ITU-T recommendation T.82. Now that the newer bi-level image compression standard JBIG2 has been released, JBIG is also known as JBIG1. JBIG was designed for compression of binary images, particularly for faxes, but can also be used on other images. In most situations JBIG offers between a 20% and 50% increase in compression efficiency over the Fax Group 4 standard, and in some situations, it offers a 30-fold improvement. JBIG uses a form of arithmetic coding patented by IBM known as the Q-coder. It bases the probabilities of each bit on the previous bits and the previous lines of the picture. In order to allow compressing and decompressing images in scanning order, it does not reference future bits. JBIG also supports progressive transmission with small (around 5%) overheads.

4.3 Overview of Image Compression

A typical image compression system is shown in Figure 4.1. It consists of three closely connected components namely: Source Encoder, Quantizer, and Entropy Encoder. Compression is accomplished by applying a linear transform to decorrelate the image

data, quantizing the resulting transform coefficients, and entropy coding the quantized values.

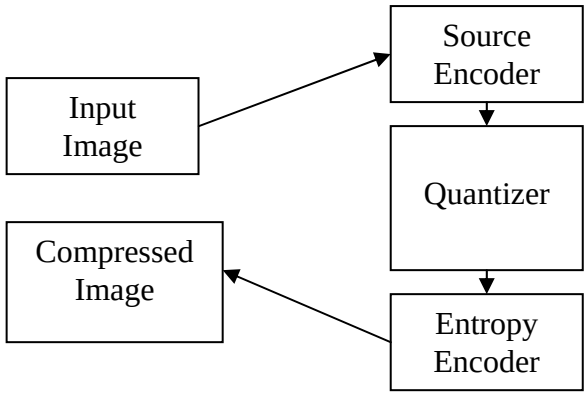

Figure 4.1 A typical image compression system

4.3.1 Source Encoder

A variety of linear transforms have been developed which include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) [84], Discrete Wavelet Transform (DWT) [85] and many more, each with its own advantages and disadvantages. DCT is the one that is used in JPEG [86]. In JPEG2000, DWT is used as source encoder. A comparison of DWT and DCT is given in 4.5

4.3.2 Quantizer

A quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ). Quantization can also be performed on a group of coefficients together, and this is known as Vector Quantization (VQ) [87].

4.3.3 Entropy Encoder

An entropy encoder further compresses the quantized values lossless to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the arithmetic encoder, although for applications requiring fast execution, simple run-length encoding (RLE) has proven very effective. An overview on various entropy encoding techniques can be found in [88].

It is important to note that a properly designed quantizer and entropy encoder are absolutely necessary along with optimum signal transformation to get the best possible compression.

4.4. An Image Compression System: JPEG 2000

Given that the most recently standardized compression system is JPEG2000, this section is dedicated to details of this compression technique.

In the baseline mode, the image is divided in 8x8 blocks and each of these is transformed with the Discrete Wavelet Transform (DWT). The transformed blocks coefficients are quantized with a uniform scalar quantizer, zig-zag scanned and entropy coded with Huffman coding. The quantization step size for each of the 64 DWT coefficients is specified in a quantization table, which remains the same for all blocks. The DC coefficients of all blocks are coded separately, using a predictive scheme. The discrete transform is first applied on the source image data. The transform coefficients are then quantized and entropy coded, before forming the output codestream (bitstream). The decoder is the reverse of the encoder. The codestream is first entropy decoded, dequantized and inverse discrete transformed, thus resulting in the reconstructed image data.

Before proceeding with each block of encoder, a process named tiling is applied to the input image. The term ‘tiling’ refers to the partition of the original (source) image into

rectangular non-overlapping blocks (tiles), which are compressed independently, as though they were entirely distinct images. Prior to computation of the forward discrete wavelet transform (DWT) on each image tile, all samples of the image tile component are DC level shifted by subtracting the same quantity (i.e. the component depth).

The authors recapitulated the encoding procedure as follows:

- The image and its components are decomposed into rectangular tiles. The tile-component is the basic unit of the original or reconstructed image.
- The wavelet transform is applied on each tile. The tile is decomposed in different resolution levels.
- These decomposition levels are made up of sub bands of coefficients that describe the frequency characteristics of local areas (rather than across the entire tile-component) of the tile component.
- The sub bands of coefficients are quantized and collected into rectangular arrays of “codeblocks”.
- The bit-planes of the coefficients in a “code-block” are entropy coded.
- The encoding can be done in such a way, so that certain regions of interest can be coded in a higher quality than the background.
- Markers are added in the bitstream to allow error resilience.
- The codestream has a main header at the beginning that describes the original image and the various decomposition and coding styles that are used to locate, extract, decode and reconstruct the image with the desired resolution, fidelity, region of interest and other characteristics.
- The optional file format describes the meaning of the image and its components in the context of the application.

The artifacts of JPEG 2000 look different from those of JPEG, have a slighter effect on the image and take higher compression levels to be visible. Often a photographic image can be compressed to 1/20 of its original (uncompressed bitmap) size without incurring visible artifacts. When the artifacts do appear, they can be seen as smoothing rather than squares or mosquito noise. The image to the right demonstrates the effects of JPEG 2000 compression in various ratios.

- The basis for JPEG´s lossy compression is two-dimensional DCT.
- The image is broken into 8 x 8 blocks on which the transform is computed.
- Image compression is obtained through quantization of these DCT coefficients to a small set of values.
- Values are entropy coded and stored as a compressed version of the image.

4.5 A Comparison of Image Encoding Quality: JPEG vs. JPEG2000

Ebrahimi et al. compared the two compression systems in [89]. Their work showed different artifacts of these compression systems which are blockiness and blur as they are described below.

Blockiness is a perceptual measure of the block structure that is common to all block-DCT based image and video compression techniques, as for example JPEG. The DCT is typically performed on 8×8 pixel blocks in the frame; the coefficients in each block are quantized separately. This leads to artificial horizontal and vertical borders between these blocks, which can be detected. Blockiness can also be caused by transmission errors, which often affect entire slices of blocks in an image.

Blur is a perceptual measure of the loss of fine detail and the smearing of edges. It is due to the attenuation of high frequencies at some stage of the recording or encoding process. It is one of the main artifacts of wavelet based compression techniques such as JPEG2000, for which transmission errors or packet loss can also induce blur. DCT-based compression schemes exhibit blur too, even if it is not the primary distortion. Other important sources of blur are low-pass filtering or out-of-focus shots. Blockiness and blur are visible in the JPEG- and JPEG2000 encoded images, respectively.

Figure 4.2 (a) Original image (b) JPEG-encoded image (c)JPEG2000-encoded image

The original image, JPEG compressed and JPEG2000 compressed images at a compression ratio of 1%100. The respective artifacts of blockiness and blur are visible in the compressed images (Figure 4.2 (b) and (c))

4.6 Conclusion

There is certain number of different compression standards all of which are widely used in many applications JPEG 2000 is standardized in year 2000 but JPEG is stil the most popular and the most used compression technique.

We have seen in this chapter the general idea behind image compression. There may be some differences between the standards but there is set of steps that do not change from one standard to other. At the end, compression is a way to represent an image. Investigation needs to be done for understanding the influence of compression on image classification.

5 EXPERIMENTATIONS

5.1 Introduction

This chapter is dedicated to the experimentations. The goals of these experimentations are: (1) to find the influence of image compression on image classification. (2) Choice of multi-class SVM method. (3) Comparison between global and local image representation for image classification.

5.2 Image Dataset Description

Our image database contains 8 categories of natural scenes: Highway, Streets, Forest, Open Country, Inside Of Cities, Tall Buildings, Coast and Mountain Images.

Figure 5.1 From top left to bottom right: Highway, Tall building, Street, Inside of city, Mountain, Open country, Coast, Forest.

The database provided by Oliva and Torralba was collected from a mixture of COREL images as well as personal photographs. All images are colored and sized of 256x256 pixels. For each classification experiment 100 images of each category are reserved for

test purpose and the remaining images are used as training set. A few sample of the image database is shown in Figure 5.1.

5.3 Description of Feature Vectors

A feature vector is a series of real numbers or integers depending on the nature of the feature. For instance if it is a histogram the entries are integers.

The SVM implementation that we used has a specific input format for learning and classification processes [90]. Examples are represented by their labels and feature vectors described in section 2. To be precise, feature vector coordinates follows the label by an index preceding each coordinate. For instance, the representation R of an image having a feature vector v of length i looks like the following:

$$R : <label> \ 1: v_1 \ 2: v_2 \ 3: v_3 \ \ldots \ i: v_i$$

An example can be labelled as positive, replacing <*label*> with the number ‘1’ or negative with ‘-1’. Positive or negative labelling is a process done at the learning step. If a classification task is being done then test examples may be labelled with number ‘0’ so that SVM knows that they are new test examples. If an example is labeled as positive and it is classified as positive then the classifier will note that it is a correct classification.

5.4 Combination of Feature Vectors

In Chapter 3 we have presented different ways of representing images like global local or intermediate representation. All of these representation approaches are feasible thanks to different sorts of features described in the same chapter as well. Each time we need to represent an image we many choices of features to use. One can prefer to represent an image by a single feature or several features at the same time by combining them. Let an image I and two feature vectors u=<u1,…um> and v=<v1,…vn> extracted from I. Then the combination of these features is the concatenation of the feature vector u v as follows:

$$W=<w_1,w_2,\ldots,w_{m+n}>=<u_1,u_2,\ldots u_n,v_1,v_2,\ldots v_m> \quad (5.1)$$

This is the most natural and easy way to combine several features. But because of the scale problem due to the nature of the individual features u and v, a normalization phase has to be introduced into the feature combination.

5.5 Feature Vector Normalization

To improve discrimination power of the image representation features can be used together. A common approach is to group the features representing an image in a single vector and then this feature vector is used as input to a single SVM classifier for obtaining the decision at once. This is the most natural and easy way to combine several features. But because of the scale problem due to the nature of the individual features a normalization phase has to be introduced into the feature combination process.

Table 5.1 Normalization methods

Minmax	$x'=\frac{x-\min\{x_k\}}{\max\{x_k\}-\min\{x_k\}}$
Decimal	$x'=\frac{x}{10^n}$
Z score	$x'=\frac{x-\mu(x)}{\sigma(x)}$
Median	$x'=\frac{x-median\{x_k\}}{MAD}$
Tanh	$x'=\frac{1}{2}\left[\tanh\left(0{,}01\cdot\frac{x-\mu(x)}{\sigma(x)}\right)+1\right]$

Numerous normalization techniques have been proposed. A collection of normalization methods used in this paper is summarized in Table 5.1.

Min–max normalization retains the original distribution of scores except for a scaling factor and transforms all the scores into a common range [0, 1]. This method is highly sensitive to outliers in the data used for estimation. *Decimal scaling* can be applied when the scores of different matchers are on a logarithmic scale. Both mean and Standard deviation are sensitive to outliers and, hence, *Z*-score normalization is not robust. If the input scores are not Gaussian distributed, *z*-score normalization does not retain the input distribution at the output. Median normalization technique does not retain the input distribution and does not transform the scores into a common numerical range. The tanh estimators are known as robust and efficient [91].

To evaluate the normalization techniques we combined the features in six different ways: first without normalization and then using the five normalization functions described in 5.5. Classification results show that normalization improves classification performance. When classification is performed without any normalization on feature vectors classification performance is even worse when compared to single feature classification performances shown in table 2. For instance even the lowest rate which is 92% for edge feature is higher than none normalization case with 88% of correct classification rate. For this modality the most performing normalization method is the decimal normalization.

We tested several combinations of global and local features for the modalities given previously. The results confirmed that decimal normalization was the best of the five normalization methods tested in this paper. Globally the normalization methods can be ranked in this way: Decimal, Tanh, Z-score, Median, Minmax. Based on this results decimal normalization is used in remaining experiments.

5.6 Performance Measures

An important aspect of our study is the use of performance criteria to evaluate multiclass SVM methods. Classification techniques are now used in many domains, and different performance metrics are appropriate for each domain. For example Precision/Recall measures are used in information retrieval. Different performance criteria measure different tradeoffs in the predictions made by a classifier, and it is possible that a learning method performs well on one metric, but be suboptimal on other metrics. Because of this it is important to evaluate algorithms on a broad set of performance metrics. There exist numerous performance measures in the literature of image classification domain. The most widely used methods are correct classification rate[92], error rate, classification accuracy in percentage [93,94] and ROC curves (sensitivity-specificity curves). Time of training is also widely used for classification techniques that require training. For SVM based classification, performance can also be evaluated by number of support vectors and the size of learned model with the other performance criteria.

Table 5.2 Confusion Matrix

		Predicted Label	
		Positive	Negative
Known Label	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Almost every performance criteria mentioned above is computed from a confusion matrix shown in Table 5.2. Accuracy is the simplest way to compare two confusion matrices because it is a measure that represents the whole classification not only one class prediction. However precision, recall and specificity are measures that represent the performance of the prediction of only one class. That is the reason why accuracy is

the most widely used measure. Kappa statistic is used to compare the degree of consensus between raters. In this context it is used to measure the quality of classification. Like accuracy, kappa statistic can represent a confusion matrix with a single value. It varies in interval [-1,1], 1 for perfect classification and -1 for a classifier that makes wrong decision systematically.

Table 5.3 Performance Measures

Measure	Formula
Precision	TP/(TP+FP)
Recall	TP/TP+FN)
Specificity	TN/(TN+FP)
Accuracy	(TP+TN)/(TP+TN+FP+FN)
F-measure	2.Precision.Recal/(Precision+Recall)
Kappa	(TP+FN)(TP+FP)/(TP+TN+FP+FN)

5.7 Influence of Image Compression on Image Classification

The next experiment's objective is to find out the influence of image compression on image classification performances. A binary classification between Forest and Highway classes is performed using texture representation. This representation contains energy, entropy, homogeneity and inertia attributes calculated from gray level co-occurrence matrices. The main idea of the experiment is to repeat the same experiment changing each time the compression ratio only. So that results of classification can be compared for different compression ratios. First, images are not much compressed; the visual quality is kept unchanged. Then the compression is augmented, even degrading visual quality.

All training and test images used in this experiment are compressed by different compression ratios (0.5, 0.2, 0.1, 0.05, and 0.01) and the experiment is conducted using the compressed images. Results are stored for each compression ratio. Samples of compressed images are shown in figure 5.3. Classification results are given in Table 5.4.

Table 5.4 Classification of compressed images

Compression Rates	1	0.5	0.2	0.1	0.05	0.01
Classification Accuracy	95.5	95.5	95.5	94.5	94	93

According to the obtained results classification performance remained unchanged for 0.5 and 0.2. Confusion matrix for the rest of the compression values are given in Tables 5.7 to 5.9. According to these results, even if the visual quality is degraded classification rates do not decrease much. The experiment has stopped at 1% compression ratio. Continuing the experiment with further compression would be unnecessary. We have demonstrated that when compression is not degrading visual quality it cannot possibly degrade the classification performance.

5.8 Choice of Representation

5.8.1 Local Representation

Local features used are color histograms, 72 bins edge direction histograms and gray level co-occurences matrices computed on blocks of 32x 32 pixels. We compared local features for four different modalities. For tall building-inside city and mountain-street modalities local representation hardly reached 80% correct classification rate.

Table 5.5 Classification accuracy for local features

Modality	**Color**	**Edge**	**Texture**
Tallbuilding/Insidecity	79%	73,5%	72%
Mountain/Street	77%	76%	80%
Forest/Highway	94%	92%	96%
Opencountry/Street	86%	93%	93%
Mean Values	84%	83,6%	85,2%

For the other modalities (forest-highway and open country-street) local representation gave satisfactory results for all the features reaching 96% of correct classification rate. Results are shown in Table 5.5. Texture and edge give better results when classification is performed between two classes for which one contains images with an important amount of texture like foliage in forest images. That was the case for forest-highway and open country-street classification. Overall texture is the most performing feature. Classification based on color or edges are as efficient.

5.8.2 Global Representation

Color, texture and gist have been compared. For extracting texture information energies of Daubechies-4 wavelet transformation applied to the LL component of the image in each step of 4 steps has been used. Color information is measured using 64 bins RGB histogram. Classification performances are given in Table 5.6.

Gist outperformed color and texture in every case. The explanation may be that gist is already a combination of basic features (naturalness, openness, roughness, expansion and ruggedness). Variations in the performance of gist are in accordance with spectral signatures that were proposed in [28].

Table 5.6 Classification accuracy for global features

Modality	**Gist**	**Color**	**Texture**
Tallbuilding/Insidecity	86,5%	71%	67%
Mountain/Street	95%	75,5%	77%
Forest/Highway	93%	90%	90%
Opencountry/Street	92%	75%	83%
Mean Values	91,6%	77,7%	79,25

5.8.3 Which Representation: Local or Global?

In this experimentation the objective is to make a choice between the two image representation approaches.

Table 5.7 Classification acuracy for the combination of local and three global features.

Modality	Local	Global
Tallbuilding/Insidecity	83	89,5
Mountain/Street	93	96,5
Opencountry/Street	96	94
Forest/Highway	97.5	94

We conducted binary classifications between different categories based on a combination of local features (color, edge, texture), and then global features (gist, color, texture).

For forest/highway and open country/street modalities the local representation outperforms the global representation while for the remaining modalities the global representation is more efficient. This can be explained by high amount of texture information contained in forest and open country images. These results are in accordance with knowledge on human vision. It suggests that when one has no information about image content both global and local information should be used for scene categorization.

5.9 Choice of Multi-class SVM Method

SVM is basically conceived for binary classification. The idea is to separate two classes by calculating the maximum margin hyperplane between the training examples. Several methods have been proposed to extend SVM in order to classify more than two classes. Currently there are two major approaches for extending SVM to multiclass classification: (1) combining several binary SVM classifiers; (2) considering all data in a single optimization. Generally the first approach is called ‘divide-and-combine’ and the second ‘all-in-one’. The main methods for divide-and-combine are One-Against-

All, One-Against-One and Directed Acyclic Graph. There is few work in the literature comparing these methods. In [93] divide-and-combine methods are compared for natural images like grass, leaves, sky etc. They conclude that OAO is most performing in terms of accuracy. In [93] satellite images (water, construction, wood, bare soil etc.) are classified using divide-and-combine methods, according to their results OAA is more performing then DAG in terms of accuracy. These comparisons do not cover all-in-one approaches and the image databases used in these work do not contain natural scenes. In [94], SVM all-in-one approach is compared to other classification techniques like neural networks, discriminant analysis and decision trees on database containing land cover images. They show that all-in-one SVM outperforms other techniques. Despite these studies there is not a fully complete comparison of multi-class SVM classification methods for natural image classification purpose. We evaluate and compare several multiclass SVM methods by following an experimental approach. We compare performance of the methods for natural image categorization task using different types of image representations. We perform an extensive evaluation using a multitude of performance measure.

5.9.1 Choice of Modalities

We use our image database to obtain two groups of images that contain both four classes. These groups are arranged in such a way that one group contain the four most similar classes and the other the four least similar ones. We use these two groups in the remaining experimentations to compare multiclass classification methods. For selection of the similar and dissimilar classes we have inspired of [28] where authors calculated the spectral signature of natural image categories. We performed binary classifications between every possible pair of classes in our image database (C(8,2)=28) based on a texture feature that contains four attributes namely energy, entropy, homogeneity and inertia extracted from gray level co-occurrence matrices. We sorted the binary classification results by accuracy. Keeping in mind that 6 classifiers are needed to build a 4-class classifier the 6 best performing classifiers sufficient to construct a 4-class classifier are selected; these four classes are the most similar ones according to the feature that is used. Following the same procedure the four least similar classes are

found. Our texture measure describes the image database in a similar way as it is described by spectral signatures. The four most similar classes are Inside of city, Street, Tall building and Mountain and the four least similar classes are Forest, Highway, Cost, Street. It has been seen that spectral signature and our measure are in accordance. We have performed binary classification of classes that belong to most similar classes between each other and the least similar classes between each other. Results are given in Table 5.8 and Table 5.9.

Table 5.8 Classification accuracy (1)Forest, (2)Highway, (3)Coast, (4)Street

1vs2	**1vs3**	**1vs4**	**2vs3**	**2vs4**	**3vs4**
95.5	94.5	87.5	78	93.5	93.5

Table 5.9 Classification accuracy (1)Inside of city (2)Street, (3)Tall building, (4)Mountain

1vs2	**1vs3**	**1vs4**	**2vs3**	**2vs4**	**3vs4**
76.5	76.5	76	92	86	71

5.9.2 Classification Based on Texture

Texture feature that is previously defined is extracted from images on blocks of 64x64 pixels. The most similar image classes are used (A) for classification. Classification results in terms of three performance measure are presented in Table 5.10. OAO with MaxWins strategy is noted as MAXWINS and OAO pairwise coupling is noted as PWC.

All three performance measures agree that the methods are ascendant ordered as DAG, PWC, MaxWins, OAA, AIO. Correlation coefficients of the performance measures are given Table 5. It has been observed that the best performing two strategies (OAA and

AIO) have similar training phases. The most discriminating performance measure is Kappa statistic because it has a wider range comparing to the others.

Table 5.10 Classification results. (A) Forest-Highway-Coast-Street.

1		**Texture**		
		Fm	*Acc*	*Kappa*
A	DAG	0.767	*0.767*	*0.690*
	PWC	0.774	*0.775*	*0.700*
	MAXWINS	0.779	*0.780*	*0.706*
	OAA	0.783	*0.785*	*0.713*
	AIO	**0.789**	***0.790***	***0.720***

Table 5.11 Classification results. (B) Inside of city-Street-Tall building-Mountain.

2		**Texture**		
		Fm	*Acc*	*Kappa*
B	DAG	0.579	0.582	0.443
	PWC	0.592	0.590	0.453
	MAXWINS	0.603	0.600	0.466
	OAA	0.552	0.587	0.450
	AIO	**0.598**	***0.605***	***0.473***

The most similar classes have been classified using all the methods based on the texture measure; results are shown in Table 5.11. The best performing method is All-In-One with agreement of the three performance measures and the worst is DAG according to Accuracy and Kappa statistic. F-measure disagreed on that decision. This is due to two classes (Inside of city and mountain) to have surprisingly low recall values (0.27 and 0.34) that decreased the mean F-measure for OAA classification. Accuracy has not been influenced by that because the other two classes (street and tall building) have very high recall values (0.88 and 0.86) that balanced low values.

5.9.3 Classification Based on Gist

The most similar classes have been classified using gist feature; results are shown in Table 5.12. The methods are ascendant ordered by their performance as DAG, PWC, OAA, MaxWins and AIO with agreement of all three performance measure. An increase of performance is observed for all five methods comparing to the classification based on texture feature with the same classes. This shows that gist is a more discriminative then texture feature.

Table 5.12 Classification results. (A) Forest-Highway-Coast-Street.

3		**Gist**		
		Fm	*Acc*	*Kappa*
A	DAG	0.874	0.875	0.833
	PWC	0.891	0.892	0.856
	MAXWINS	0.914	0.915	0.886
	OAA	0.904	0.905	0.873
	AIO	**0.941**	***0.942***	***0.923***

Table 5.13 Classification results (B) Inside of city-Street-Tall building-Mountain

4		Gist		
		Fm	*Acc*	*Kappa*
B	DAG	0.773	0.775	0.700
	PWC	0.816	0.817	0.756
	MAXWINS	0.819	0.820	0.760
	OAA	0.845	0.847	0.796
	AIO	**0.858**	***0.860***	***0.813***

Classification results of the four most similar classes using gist feature is shown in Table 5.13. Methods are ordered in ascendant rank as: DAG, PWC, MaxWins, OAA, AIO with agreement of all the performance criteria. DAG, PWC and MaxWins that have the same binary classifiers are grouped together in performance rank. OAA and AIO made a second group with very similar results that can be explained by the similarity of their training phases.

Table 5.14 Correlation Coefficients

	Fm-Acc	**Fm-Kappa**	**Acc-Kappa**
1	0.9983	0.9987	0.9995
2	0.6786	0.6581	0.9995
3	1.0000	0.9999	1.0000
4	0.9999	1.0000	1.0000

Table 5.14 shows the correlation coefficients of performance measures. Correlation coefficients vary in interval [-1,1]. 1 is for perfect agreement between measures and -1 for disagreement. There is perfect agreement between performance measures for all experiments except from the second experiment for which the reason of the disagreement is explained in 5.9.2.

5.10 Conclusion

Results show that All-In-One method is the most performing SVM multiclass classification strategy for natural scene classification. This conclusion is confirmed with all four experiments performed on two separate groups of images using two different types of representation, one local, one global. Three of four experiment conclusions agreed that One-Against-All is the second best performing method. This result is interesting because OAA and AIO methods have very similar training phases where one class is considered against the rest of the classes.

6 CONCLUSION

Image categorization involves several contributions. A very large range of research area contributed to image categorization work. Cognitive sciences, Statistics and Computer sciences are some of these areas. Cognitive Sciences have especially an important role in image categorization works because categorizing an image is obviously a task best performed by human. Therefore researchers working in this area need a good knowledge and understanding of human visual system and human perception in order to emulate it with computational system. Statistics has also a very important role because of the classification problem that requires statistical techniques and finally computational sciences are inevitable for design and implementation of such a system. This thesis can be classified as a study in both computational domain and statistics.

We have surveyed different classification techniques and algorithms besides Support Vector Machines, some of these techniques that we have surveyed are: Artificial Neural Networks, Naive Bayes Classifier, Hidden Markov Model and K-Nearest Neighbor. Among all these techniques Support Vector Machines was the most promising. It is relatively a new technique and it is very popular in this domain. We have concentrated on SVMs as a tool for image categorization objective. We has surveyed multi-class classification approaches using SVM in detail.

Image representation was another focus of this work. In fact representation of images has certainly the most important role in image categorization. Categorization is possible only if images can be discriminated. So we need discriminative image characteristics to achieve this goal. We have reviewed different kinds of image representations namely local, global and intermediate representations. We have proposed image compression as an intermediate image representation.

Finally, we have conducted a series of experiments for different objectives which are: understanding the influence of image compression in image classification, comparing local and global representation and comparing multi-class SVM strategies.

References

[1] Webb, A., *Statistical Pattern Recognition,* Wiley, (2002).

[2] Zhang, H., *The Optimality of Naive Bayes*, Wiley, (2001).

[3] Vailaya, A., Jain, A., Zhang, H.J., "On Image Classification: City Images vs. Landscapes", Pattern Recognition Journal, Vol. 21, 3-8, (2002).

[4] Sebe, N., Cohen, I., Garg, A., Huang, S., *Machine Learning in Computer Vision*, Springer, (2005).

[5] Li, J., Wang, J.Z., "Automatic Linguistic Indexing of Pictures by a Statistical Modelling Approach", IEEE Trans. on PAMI 25, 1075-1088, (2003).

[6] Mahmoud, S., Moumen T., Melegy, E., "Statistical and Neural Methods for Remote-Sensing Image Classification and Decision Fusion: A Comparative Study", Electrical, Electronic and Computer Engineering, ICEEC '04. (2004).

[7] Vogel, j., Schiele, B., "A Semantic Typicality Measure for Natural Scene Categorization", DAGM 2004, LNCS 3175, 195-203, (2004).

[8] Bishop C.M., *Neural Networks for Pattern Recognition*, Oxford (2005).

[9] Pontil, M., Verri, A. "Support Vector Machines for 3-d Object Recognition", Pattern Anal. Machine Intell., 20, (1998).

[10] Blanz, V., Schfolkopf, B., Burges C., Vapnik, V., Vetter, T., “Comparison of View-based Object Recognition Algorithms Using Realistic 3d Models”, Artificial Neural Networks, ICANN’96, Berlin, Germany, 251-256, (1996).

[11] Christopher, J.C., “A Tutorial on Suppport Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 121-167, (1998).

[12] Busuttil, S., "Support Vector Machines with Profile-Based Kernels for Remote Protein Homology Detection”, http://citeseer.ist.psu.edu/732559.html (1999).

[13] Platt, J.C., Christianini, N., Shawe-Taylor, J., “Large Margin DAGs for Multiclass Classification”, Advances in Neural Information Processing Systems, Vol. 12 547-553, (2000).

[14] Li, X., Wcing, L., Sung, E., “Multi-label SVM Active Learning for Image Classification”, International Conference on Image Processing (ICIP’04), (2004).

[15] Chen, Y., Wang, J. Z., “Image Categorization by Learning and Reasoning with Regions”, Journal of Machine Learning Research 5, 913-939, (2004).

[16] Tsai, C.F., Mcgarry, K., Tait, J. “CLAIRE: A Modular Support Vector Image Indexing and Classification System” ACM Transactions on Information Systems, 24(3), 353-379, (2006).

[17] Williamowsky, J., Arregui, D., Csurka, G., Christopher, R., Fan, L. “Categorizing Nine Visual Classes Using Local Appearance Descriptors”, ICPR Workshop Learning for Adaptable Visual Systems Cambridge, (2001).

[18] Fan, J., Gao, Y., Luo, H., Xu, G., “Statistical Modeling and Conceptualization of Natural Images”, Pattern Recognition 38, 865-885, (2005).

[19] Chapelle, O., Haffner, P., Vapnik, V., “Support Vector Machines for Histogram-Based Image Classification”, IEEE Trans., on Neural Networks 10, 1055-1064, (1999).

[20] Krebe1, U., “Pairwise Classification and Support Vector Machines”, Advance in Kernel Methods Support Vector Learning, 255-268, (1999).

[21] Gao, Y., Fan J., “Semantic Image Classification with Hierarchical Feature Subset Selection”, MIR’05, November 10–11, (2005).

[22] Zhu, Q., Yeh, M.C., Cheng, K.T. “Multimodal Fusion Using Learned Text Concepts for Image Categorization”, MM'06, USA, (2006).

[23] Vapnik V., *Statistical Learning Theory*, Wiley, (1998).

[24] Weston J., Watkins, C., *Multi-Class Support Vector Machines*, Royal Holloway, (1998).

[25] Hsu C.W., Lin, C.J., “A Comparison of Methods for Multi-class Support Vector Machines”, IEEE Trans. Neural Networks, 13(2), 415-425, (2002).

[26] Barla, A., Odone, F., Veri, A., “Old Fashioned State-of-the-art Image Classification”, 12th International Conference on Image Analysis and Processing (ICIAP’03), (2003).

[27] Guan, X., Pan, G., Wu, Z., “Automatic Categorization of Traditional Chinese Painting Images with Statistical Gabor Feature and Color Feature”, ICCS 2005, LNCS 3514, 743-750, (2005).

[28] Oliva, A., Torralba, A., “Modeling The Shape of The Scene: A Holistic Representation of The Spatial Envelope”, Int. Journal of Computer Vision 42, (2001).

[29] Barnard, K., Forsyth, D., “Learning the Semantics of Words and Pictures”, Proc. Int. Conf. Computer Vision 2, 408-415, (2001).

[30] Szummer, M., Picard, R., “Indoor-outdoor image classification”, Int. Workshop on Content-based Access of Image and Vedeo Databases, Bombay, India, (1998).

[31] Barrow, H.G., Tannenbaum, J.M., “Recovering Intrinsic Scene Characteristics from Images”, Computer Vision Systems, A. Hanson and E. Riseman (Eds.), Academic Press, New York, 3-26, (2005).

[32] Marr, D., *Vision*, WH Freeman: San Francisco, CA. (1982).

[33] Biederman, I., “Aspects and Extension of a Theory of Human Image Understanding”, Computational Processes in Human Vision: An Interdisciplinary Perspective, Z. Pylyshyn (Ed.), Ablex Publishing Corporation, (1988).

[34] Potter, M.C., “Short-Term Conceptual Memory for Pictures”, J. Exp. Psychol, Vol. 2, 509-522, (1976).

[35] Schyns, P.G., Oliva, A., “From Blobs to Boundary Edges: Evidence for Time- and Spatial-Scale-Dependent Scene Recognition”, Psychol. Sci., Vol.5, 195-200, (1994).

[36] Thorpe, S., Fize, D., Marlot, C., “Speed of Processing in the Human Visual System”, Nature, Vol. 381 520-522, (1996).

[37] Torralba, A., Oliva, A., “Building the Gist of a Scene: The Role of Global Image Features in Recognition”, Progress in Brain Research (2006).

[38] Potter, M.C. “Meaning in visual scenes”, Science, Vol. 187, 965-966, (1975).

[39] Potter, M.C., Staub, A., O’ Connor, D.H., “Pictorial and Conceptual Representation of Glimpsed Pictures”, J. Exp. Psychol. Hum. Percept. Perform., (2004).

[40] Oliva, A., “Gist of the Scene”, Neurobiology of Attention, Elsevier, San Diego, CA 251-256, (2005).

[41] Navon, D., “Forest Before Trees: the Precedence of Global Features in Visual Perception”, Cognit. Psychol., Vol. 9, 353-383, (1977).

[42] Kimchi, R., “Primacy of Holistic Processing and Global/local Paradigm: A Critical Review”, Psychol. Bull., Vol. 112, 24-38, (1992).

[43] Kimchi, R., “Uniform Connectedness and Grouping in the Perceptual Organization of Hierarchical Patterns”, J. Exp. Psychol. Hum. Percept. Perform., Vol. 24, 1105-1118, (1998).

[44] Renninger, W., Malik, J., “When is Scene Identification Just Texture Recognition?”, Vision Research, 2301-2311, (2004).

[45] Friedman, A., “Framing Pictures: The role of Knowledge in Automatized Encoding and Memory for Gist”, Journal of Experimental Psychology: General, Vol.108, 316-355, (1979).

[46] Biederman, I., “Perceiving Real-World Scenes”, Science, 177, 77–80, (1972).

[47] Loftus, G.R., Nelson, W.W., Kallman, H.J., “Differential Acquisition Rates for Different Types of Information from Pictures”, Quarterly Journal of Experimental Psychology: Human Experimental Psychology, 35, 187–198, (1983).

[48] Metzger, R.L., Antes, J.R., “The nature of Processing Early in Picture Perception”, Psychological Research, 45(3), 267–274, (1983).

[49] Fei-Fei, L., Perona, P., “A Bayesian Hierarchical Model for Learning Natural Scene Categories”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 524, (2005).

[50] Vailaya, A., Figueiredo, M., Jain, A., Zhang, H., “Image Classification for Content-based Indexing”, IEEE Trans. on Image Processing, 10, (2001).

[51] Bosch, A., Zisserman, A., Munoz, X., “Scene Classification via pLSA”, Proceedings of the ECCV, (2006).

[52] Swain, M.J., Ballard, D.H., “Color Indexing”, Int. J. of Computer Vision, Vol. 7, 11-22, (1991).

[53] Gong, Y., Proietti, G., Faloutsos, C., “Image Indexing and Retrieval Based on Human Perceptual Color Clustering”, Proc. of International Conference on Computer Vision and Pattern Recognition(CVPR), June (1998).

[54] Stricker and A. Dimai. “Color Indexing with Weak Spatial Constraints”. SPIE Proc., Vol. 2670, 29 - 40, (1996).

[55] Pass, R., Zabih, T., “Histogram Refinement for Content Based Image Retrieval”, Proc. of the Third IEEE Workshop on Applications of Computer Vision, Sarasota, (1996).

[56] Paschos, G., Radev, I., “Image Content-Based Retrieval Using Chromaticity Moments”, IEEE Transactions On Knowledge And Data Engineering, 15(5), (2003).

[57] Shih, J.L., Chen L.H., “Colour Image Retrieval Based on Primitives of Colour Moments”, IEEE Proc-Vis Image Signal Process. Vol. 149 No 6 December (2002).

[58] Huang, J., Kumar, S., Mitra, M., Zhu W.J., Zabih, R., “Image Indexing Using Color Correlograms”, Proc. of Conf. on Comp. Vision and Patt. Recog.(CVPR), San Juan (Puerto Rico), 762-768, (1997).

[59] Medioni, G., Kang, S.B., “Emerging Topics in Computer Vision”, IMSC Multimedia Series, Vol. 8, 212-2120, (2005).

[60] Mallat, S., Clerk M., “The Texture Gradient Equation for Recovering Shape from Texture”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(4), 536-549, (2002).

[61] ISO/IEC JTC1/SC29/WG11/N4063, “MPEG-7 Visual part of Experimentation Model Version 10.0”, Singapore, March (2001).

[62] Canny, A., “A Computational Approach to Edge Detection”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6), 679-698, (1986).

[63] Abdel-Mottaleb, M., “Image Retrieval Based on Edge Representation”, Proc. Int. Conf. on Image processing, Piscatway, NJ, USA, Vol. 3, 734–737, (2000).

[64] Jain, K., Vailaya, A., “Image Retrieval Using Color and Shape”, Pattern Recognition, 29(8), 1233-1244, (1996).

[65] Shih, J.L., Chen, L.H., “A New System for Trademark Segmentation and Retrieval”, Image Vis. Comput., 19(10), 1011-1018, (2001).

[66] Yoo, H.W., Jang, D.S., Jung, S.H., Park, J.H., Song, K.S., “Visual Information Retrieval System Via Content-Based Approach”, Pattern Recognit., 35(3), 749–769, (2002).

[67] Zhu, Q., Chen, M., Cheng, K.T., “Multimodal Fusion Using Learned Text Concepts for Image Categorization**”,** Proceedings of the 14th annual ACM international Conference on Multimedia 211-220, (2006).

[68] Torralba, A., Oliva, A., “Semantic Organization of Scenes Using Discriminant Structural Templates”, Proceedings of International Conference on Computer Vision, ICCV99, Korfu, Grece, 1253-1258, (1999).

[69] Lowe, D., “Distinctive Image Features from Scale-Invariant Keypoints”, IJCV, 60(2), 91-110, (2004).

[70] D. Blei, A., Jordan, M., "Latent Dirichlet Allocation", Journal of Machine Learning Research, 993-1022, (2003).

[71] Fei-Fei, L., Fergus, R., Torralba, A., “Recognizing and Learning Object Categories”, http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html, (2007).

[72] Barnard, K., Duygulu, P., Guru, R., Gabbur, P., Forsyth, D., “The Effects of Segmentation and Feature Choice in a Translation Model of Object Recognition”, IEEE CVPR, 675-682, (2003).

[73] Sivic, J., Zisserman, A., “Efficient Visual Content Retrieval and Mining in Videos”, Pacific-Rim Conference on Multimedia, (PCM 2004), Tokyo, Japan, (2004).

[74] Winn, J., Criminisi, A., Minka, T., “Object Categorization by Learned Universal Visual Dictionary”, In Proc. ICCV, Beijing, China, (2005).

[75] Lowe, D., "Object Recognition with Informative Features and Linear Classification", Proc. of International Conference on Computer Vision, 1150-1157. (1999).

[76] Lazic, N., Aarabi, P., “Importance Of Feature Locations In Bag-Of-Words Image Classification”, IEEE ICASSP, (2007).

[77] Mikolajczyk, K., Schmid, C., “A Performance Evaluation of Local Descriptors”, IEEE Trans. on PAMI, Vol. 27, 1615- 1630, (2005).

[78] Wallace, G.K., “The JPEG Still Picture Compression Standard”, Communication of the ACM, 34(4), 30-44, (1991).

[79] Carpentieri, B., J.Weinberger, M., Seroussi, G., “Lossless Compression of Continuous-Tone Images”, Proceedings of IEEE, 88(11), 1797-1807, (2000).

[80] Christopoulos, C., Skodras, A., Ebrahimi, T., “The JPEG2000 Still Image Coding System: An Overview”, IEEE Trans. on Consumer Electronics, 46(4), 1103-1127 (2000).

[81] Welch, T.A., “A Technique for High-Performance Data Compression”, Computer, 17(6), 8-19, (1984).

[82] PNG (Portable Network Graphics) Specifications, http://www.w3.org/TR/PNG/#4Concepts.Sourceimage, (2007).

[83] Santa-Cruz, D., Grosbois, R., Ebrahimi, T., “JPEG 2000 Performance Evaluation and Assessment”, Signal Processing: Image Communication, 113-130, (2002).

[84] Ahmed, N., Natarajan, T., Rao, K.R., “Discrete Cosine Transform”, IEEE Trans. Computers, Vol. 23, 90-93, (1974).

[85] Vetterli, M., Kovacevic, J., “Wavelets and Subband Coding”, Englewood Cliffs, NJ, Prentice Hall, (1995).

[86] JPEG (Joint Photographic Experts Group) http://www.jpeg.org/jpeg/index.html, (2007).

[87] Gersho, A. Gray, R.M., *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, (1991).

[88] Nelson, M., *The Data Compression Book*, 2nd ed., M&T books, (1995).

[89] Ebrahimi, F., Chamik, M., Winkler, S., “JPEG vs. JPEG2000: An Objective Comparison of Image Encoding Quality”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2003).

[90] SVM-Light Support Vector Machine http://svmlight.joachims.org, (2007).

[91] Jaina, A., Nandakumara, K., Rossb, A., “Score Normalization in Multimodal Biometric Systems”, Pattern Recognition 38, 2270-2285, (2005).

[92] Foody, G.M., Mathur, A., **“**A Relative Evaluation of Multiclass Image Classification by Support Vector Machines”, Geoscience and Remote Sensing, IEEE Transactions, 42(6), 1335-1343, (2004).

[93] Ren, J., Shen, Y., Ma, S., “Applying Multi-class SVMs into Scene Image Classification”, Proceedings of the 17th international conference on Innovations in applied artificial intelligence, 924-934, (2004).

[94] He, L., Kong, F., Shen**,** Z., **“**Multiclass SVM Based Land Cover Classification with Multisource Data”, Proceedings of International Conference of Machine Learning and Cybernetics, 6(18), 3541-3545, (2005).

Biographical Sketch

Can Demirkesen was born in Istanbul, Turkey on October 16, 1981. He graduated from Galatasaray High School in 2000. He received his B.S. degree in Computer Engineering in 2005 from Galatasaray University, Istanbul, Turkey.