Image Annotation and Content Based Image Retrieval centered Image Query System

Karan Sikka and Nitesh Sinha[1]

Under the guidance of Dr. Kannan Karthik[2]

Department of Electronics & Communication Engineering
Indian Institute of Technology Guwahati

May 2010

A thesis submitted to the Indian Institute of Technology Guwahati in partial fulfillment of the requirements for the degree of Bachelor of Technology.

Certificate

This is to certify that this thesis entitled **"Image Annotation and Content Based Image Retrieval centered Image Query System"** is a bonafide work of **Karan Sikka (Roll No. 06010219) and Nitesh Sinha (Roll No. 06010227)**, carried out in the Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, under my supervision and that it has not been submitted elsewhere for a degree.

Dr. Kannan Karthik
Assistant Professor
Department of Electronics and Communication Engineering
Indian Institute of Technology Guwahati
May 3, 2010

[1]06010219 and 06010227, k.sikka,nitesh@iitg.ernet.in
[2]k.karthik@iitg.ernet.in

Abstract

Image retrieval has gained a significant importance for searching and locating images either encompassing common elements or belonging to the same class. With a massive increase in the digital data, the process to find best matches for a query image manually using human subjetivity has become a cumbersome task. The abilty of machines to match the subjective analysis of humans lies at the core of images retrieval and renders the task extremely complex.

As part of our bachelor's thesis project, a system has been designed for image retrieval addressing both, the importance of low-level visual features and semantic (subjective) information. In order to enhance the celerity of the process, the levels of accuracy and options available with the user, the system works both automatically and semi-automatically. The algorithm employs an automatic segmentation algorithm since a region based query is more closer to human visual perception. The proposed system follows a two step approach. The first step being the fuzzy association of a query (or regions) with multiple semantic labels. This approach is based on the mapping of different concepts on the visual space using Gaussian mixture model (GMM) training approach using maximum likelihood (ML). This fuzzy annotation is employed to extract the top results from the image database. The subsequent step uses the subset of the images generated from the previous process and implements a host of region based features like colour, texture, etc for estimating a refined-similarity ranking of the database images.

The current formulation is equipped with many advantages. Firstly, the semantic modelling takes into account the fact that similar objects can exist with different visual features and vice-versa. Secondly, the inclusion of context is of paramount importance in image retrieval task since it reduces the ambiguity owing to correlation in image features for different objects. This fusion of this formulation with region based query further improves the results.

The algorithm has been evaluated against a standard repository, known as Wang database, consisting of 800 images belonging to widely disparate classes i.e. sea-beaches, elephants with trees/sand in background, horses, buses, monuments, mountains, etc. The algorithm yields satifactory results with most of the test images.

Keywords **:** Semantic information, GMM, ML

Acknowledgments

We would firstly like to thank our guide, Dr. Kannan Karthik for giving us this project. We would also like to thank him for allowing us all the latitude for doing research in the domain of the project and being generous with his valuable guidance and suggestions.

We would like to express our gratitude towards Dr. M.K Bhuyan, Assistant Professor, Dept. of Electronics and Communication Engineering, IIT Guwahati for taking the courses on image processing and computer vision. Thus helping us to build our knoweldge in these domains.

Next, we would like to thank our friend and colleague, Mr. Mukund R. (Final year B.Tech student, CSE, IIT Guwahati) for helping us with the LaTeXfor bringing this report in its current impressive format.

We would also like to thank the Department of Electronics and Communication Engineering, IIT Guwahati, for providing all the necessary assistance in carrying out this project.

Karan Sikka, Nitesh Sinha
May 3, 2010

Contents

List of Figures

Chapter 1

Introduction

With the advent of digital technologies, storing, processing and analysis of images have become quite convenient. This is also supplemented by a drastic cost reduction in image acquisition devices as a result of which digital images now play an important role in depicting and disseminating pictorial information.

Consequently, large image databases are being created and used in a number of applications, including criminal identification, multimedia encyclopaedia, geographic information systems, online respository for art works, medical image archives, and other such areas. Information retrieval forms an integral part of these archives and has been gaining importance over the years. This application can be understood in the context of certain examples. Radiologists have to access large amounts of images daily [17], home-users often have image databases of thousands of images [22], and journalists also need to search for images by various criteria [1]. Although various image compression algorithms have been proposed over the years to make these wieldy digital archives more manageable, the enormity of data has precluded their effectiveness as well. The use of textual information (meta-data) has been the only technique in the past. But, this task of associating meta-data with each image is quite cumbersome owing to its being time intensive. Increased complexity due to the incorporation of text is another disadvantage of this methodology.

Image retrieval can be done in two fundamental ways. One method is to use low level features like colour, texture, etc and middle level features like boundaries, shape, etc. Such a method has the advantage of taking into account every minute detail while calculating these feature values. A conventional content based image retrieval (CBIR) technique falls under this category. It aims at describing the complex information of digital images by non-textual features that increases the efficiency of query handling. Such systems utilize the low level features that can be extracted from an image directly or after requisite processing. These features are then used to search the database using one of the various distance metrics available for this purpose. The second category employs high level features i.e. features modelling human subjective perception. The fact that it is extremely difficult for raw image features to model the human judgement precisely, this technique inevitably involves human intervention. This fomulation involves construction of object-ontologies [16] to define high-level features, application of machine learning approach to learn high-level concepts from low level concepts [23] and relevance feedback [20].

The methods based on low-level features can utilize global features or local features for objects or regions of interest. The first major CBIR system was QBIC [4] developed by researchers from IBM. This system uses three types of features: colour histograms, moment based shape features and texture descriptors. Another popular CBIR system is Blob World system [2], developed at University of California, Berkeley. This is one of the first methods using region based query handling and is based on maximum likelihood based algorithm that clusters image pixels based on colour, texture and position information. But since this algorithm employs image segmentation algorithm, it suffers from the drawbacks of segmentation (clustering). Other famous algorithms include SIMBA [21], IRMA [10], CIRES [7] and others. But none of these techniques has been able to provide a comprehensive solution to the problem of CBIR and thus has been an area of active research.

The other fundamental method used for faithful image retrieval uses high level features which attempt to simulate human judgement (which might incorporate human intelligence for learning, etc) by incorporating semantic information like context. Relevance feedback is the most widely used technique for using human perception for retrieval and refinement of results. In this, the user selects the relevant results after each iteration in order to refine the results until a definite level. Although, there is an increase in the relevant results, this method is restricted by the visual features of relevant images. The second method uses object-ontology where labels for various regions are defined in our usual language. For example, water can be represented as 'below, textured and blue region'. This method works well for database with limited variation in the visual properties and the structure of different regions, however better technques are required for tackling databases with higher variation in individual regions. The machine learning approach aims at predicting the value of an outcome measure (semantic categories/ labels) based on a set of input measures (visual features) [11]. Howoever, such techniques also suffer from a series of problems. Firstly, it becomes practically difficult to develop a lexicon for the entire range of objects that may exist in the image. Under such circumstances, an object belonging to an undefined class gets labelled with available code-words. Secondly, if this labelling is not perfomed comprehensively then many of the smaller regions in an image gets left out. Hence, although the system performs well for larger regions, the performance is quite unsatisfactory for smaller objects existing within an image [24].

The phase-I of the current work focused on implementation of a CBIR system based on global features. In order to incorporate human perception in the retrieval process, a feedback system was also introduced wherein the user can further refine the results, thus making the system dynamic. The user conveyed his/her "opinion" by ranking the top results produced after a retrieval operation with standard weights applied to these features. The feedback information fed by the user was subsequently employed to automatically update the feature weights and re-compute the refined results. Howoever, region based image retrieval (RBIR) is more fundamental than the approach based on global features since comparison can be made at the object level. Hence, an RBIR implementation was chosen for phase-II (proposed algorithm) of this project.

As a novelty of this work, both low and high level techniques have been integrated to circumvent the shortcomings of these individual methods as discussed earlier. The intention is to synergize the positive elements from both the algorithms and improve the overall performance of the algorithm with as less com-

putational requirements as possible. Broadly, the method comprises the implementation of RBIR preceded with a system of fuzzy image annotation. The algorithm begins with segmentation of images using JSEG algorithm [3]. This is followed by training of a Gaussian mixture model (GMM) of feature vectors corresponding to various image class. The training samples consists of manually annotated regions for each category obtained after segmentation. The whole process leads to estimates of the class-conditional probablities of visual features for seventeen semantic categories or classes. Following this, all the images in the database (including the training images) are alloted fuzzy annotated by the algorithm. On feeding a query, the system annotates each of the segmented regions automatically based on the posterior probabilities. Further, information from co-occurance of differnt regions (region consistency) is employed to reduce the ambiguity existing due to overlapping feature space for different categories. As a result a region can be classified into at most two classes. The classficiation information from each region is employed to generate a annotation class probabilty vector (ACPV). This is followed by identifying and segregating all the images in the database having an ACPV similar to the query image. Finally, the ordering of the images of the subset so generated is peformed using the RBIR.

The rest of the report has been organized in the following manner. Chapter 2.1 focuses on the CBIR based on global features. Chapter 3.1 explains the algorithm that has been used for segmentation of images prior to any feature extraction and retrieval. Chapter 3.2 defines the algorithm used for annotation under the heads of training methodology, feature vectors used, annotation technique and intermediate result listing. Chapter 3.3.1 details upon the technique of RBIR along with the explanation of features used. Chapter 4 details upon the algorithm implementation environment and other details. This is followed by discussion of results in Chapter 5. Chapter 6 highlights some of the novel contributions made in this work. Chapter 7 briefs about the future work that may be done. Chapter 8 concludes the report.

Chapter 2

CBIR Based on Global Features

This chapter covers the Phase-I i.e. the work done in the initial semester during this project.

2.1 Global Features

This portion of the chapter describes the various features that have been employed in this algorithm. A detailed analysis also reveals that these features tend to convey information regarding the number of objects in an image, uniformity of the background, distribution of foreground objects etc. and thus form an integral part of any content based image retrieval system.

2.1.1 Colour Histogram Distance (CHD)

This feature was based on the histogram of the colour components present in the image. Experimentations were done with different number of bins and a 64 bin histogram was found to give best results.

An equation is implemented for calculating the colour histogram distance (CHD), represented by α, between each database image and the query image fed into the system. A metric was improvised, whereby the Euclidean distance between each bin is suppressed with each increasing bin. This strategy has been employed since most of the images (as in the current database) are marked by dominating backgrounds corresponding to higher intensity levels (or bins) in the histogram. In case the distance is not subdued towards the higher intensity levels, the system is more likely to retrieve images having similar backgrounds with an equivalent area as that of the query image. Thus a factor is required in the distance metric to continuously reduce the dominance of the higher intensity values on the final results, making the lower intensity values more prominent. Given above considerations, the following equation has been implemented for calculating $\propto$ for each of the database image,

$$\alpha = \left[\sum_{c=1}^{3} \sum_{j=1}^{64} \left[\frac{h_{qc}(j) - h_{dc}(j)}{2 + 4(j-1)} \right]^2 \right]^{\frac{1}{2}} \tag{2.1}$$

where, $h_{qc}(j)$ represents the value at j^{th} bin of the of colour c histogram of the query image and $h_{dc}(j)$ represents the value at j^{th} bin of the of colour c histogram of the database. The denominator accomplishes the task of assigning lower weights to higher bins in the histogram.

2.1.2 Downsampled Image Distance

The number of pixels in an image can be correlated to the fineness of different components present in an image. However, high resolution often makes it difficult to compare the spatial features amongst images. This can be attributed to the fact that even a minute displacement in one of the objects present in both images would render pixel by pixel comparison erroneous. At the same time, downsampling cannot be done indiscriminately because an excessive undersampling can accompany high losses in feature, causing visually different images to be classified as similar. Thus, each image has been downsampled into a size of 32x32 pixels for a pixel based comparison using Euclidian distance. Such a size best accounts both the considerations. The following equation has been implemented to calculate the downsampled image distance (DID) represented as β.

$$\beta = \sum_{c=1}^{3} \sum_{i=1}^{32} \sum_{j=1}^{32} |di_{qc}(i,j) - di_{dc}(i,j)| \tag{2.2}$$

where, $di_qc(j)$ represents the value at $(i,j)^{th}$ pixel of the downsampled query image and $di_{dc}(i,j)$ represents the value at $(i,j)^{th}$ pixel of the downsampled database.

2.1.3 Co-Occurrence Matrix Based Distances

A texture in an image is characterized by a somewhat regular distribution pattern of pixels. The smallest element in such a pattern which when repeated in all the direction produces the original texture is called a texel. This texel belongs to the class of global texture descriptors and is derived from the co-occurrence matrix as proposed by Haralick [5]. These descriptors estimate the textural aspects of an image by measuring properties like coarseness, randomness etc. of an image. Each element $M_d(i,j)$ of the gray level co-occurrence matrix contains the count of the pixels separated by displacement vector $d = (d_x, d_y)$ and having gray levels i and j. The dimension of this matrix is $n \times n$, where n is the total number of gray levels in an image. In order to allow for the comparison of images with different dimensions, this matrix is normalized before any comparisons. The normalization is achieved by employing the following equation

$$M[i,j] = \frac{M[i,j]}{\sum_i \sum_j M[i,j]} \tag{2.3}$$

Following features are extracted from this matrix

2.1.3.1 Contrast

It is a measure of the local variations present in an image and is given

$$C = \sum_i \sum_j (i-j)^2 M_d(i,j) \tag{2.4}$$

The number of pixels in each cell of the matrix is weighted by the $(i-j)^2$ term to consider the gray level difference between the i^{th} and j^{th} pixel while calculating C. In case of a uniform region, the pixels will be distributed around the diagonal, yielding a low value C and vice-versa.

When a query image is encountered that is not present in the database, the contrast value is calculated for it. Subsequently, the distance of contrast for each image in the database is calculated by using the modulus distance between the values i.e.

$$d_c = |C_q - C_d| \tag{2.5}$$

where, C_q and C_d are the contrast values of the query and a database image.

2.1.3.2 Homogeneity

This feature is complementary to contrast of an image. It calculates the level of homogeneity in an image (spread along the diagonal elements) and is formulated as

$$H = \sum_i \sum_j \frac{M_d(i,j)}{(1+|i-j|)} \tag{2.6}$$

The distance of this feature between the query and a database image is calculated as

$$d_h = |H_q - H_d| \tag{2.7}$$

where, H_q and H_d are the homogeneity values of the query and a database image.

2.1.3.3 Entropy

It is a measure of randomness in an image and is formulated as

$$E = -\sum_i \sum_j M_d(i,j) log(M_d(i,j)) \tag{2.8}$$

It measures the randomness of the texels present in an image. Thus an image possessing a number of texture regions will give a higher value of entropy. In order to visualize this measure, one can consider a case of an image with a single object in front of a uniform background. The images belonging to this class will have lower entropy than those having multiple objects in foreground.

The distance between the entropy values of the query image with those present in the database is calculated by using the modulus distance i.e.

$$d_e = |E_q - E_d| \tag{2.9}$$

where, E_q and E_d are the entropy values of the query and a database image.

2.1.3.4 Maximum Probability

This refers to the largest entry in the matrix, which corresponds to the dominant pixel pattern in an image. It is given as

$$(i_{min}, j_{min}) = \{(i', j') | M_d(i', j') \leq M_d(i, j) \forall (i, j) \neq (i', j')\} \tag{2.10}$$

This refers to the largest entry in the matrix, which corresponds to the dominant pixel pattern in an image. It is given as

$$d_{mp} = |i_{qmax} - i_{dmax}| + |j_{qmax} - j_{dmax}| \tag{2.11}$$

where, i_{qmin}, j_{qmin} and i_{dmin}, j_{dmin} correspond to the query and the database image respectively.

2.2 Global Algorithm

The final results with CBIR are generated by taking a weighted linear sum of the individual distances estimated for each feature between two images. The equation is as follows

$$d_c(I_q, I_d) = \sum_f W_{cf} d_{cf}(I_q, I_d) \tag{2.12}$$

where, W_{cf} represents the weight assigned to the feature f. $d_{cf}(I_q, I_d)$ is the normalized distance corresponding to feature f between images I_q and I_d, the query and the database images respectively. The distance $d_{cf}(I_q, I_d)$ is calculated using the distance metric defined for feature f as given in Chapter 2.1. $d_c(I_q, I_d)$ represents the total distance between the two images. Distances corresponding to each feature for a query image have been normalized in the range of 0 to 1.

Previously, a number of feedback techniques had been introduced to tackle these issues [20]. Building on the same lines, a new algorithm has been proposed for refining retrieval process based on obtaining relevance feedback from the user. The user initiates this process by interactively selecting the two best results from the outputs so produced. Based on this, the weights are updated based on the following formulation

$$W_{cf} = \frac{1}{\omega_1 d_{cf}(I_q, I_d) + \omega_2 d_{cf}(I_q, I_d)} \tag{2.13}$$

ω_1 and ω_2 define the precedence levels for the two feature distances in determining the weights. The values for these parameters has been experimentally verified to be most suited at $\omega_1 = 7000$ and $\omega_2 = 3000$.

This step updates the weights W_{cf} for each of the feature f. The contention behind using this scheme for weight adjustment is that a higher priority should be provided to feature distances having least values. The underlying logic is that if a database image is identified as visually similar to the query image, then

it will have a relatively smaller D as compared to other database images. The smaller D can be attributed to smaller $d_{cf}(I_q, I_d)$ for certain features, which are responsible for making a database image similar to the query image. Hence, it is legitimate for these feature distances to have a higher precedence while identifying similar images. The inverse relation of the feature distances of the best two results implements this logic effectively. The results with the new weights are presented to the user.

Chapter 3

Proposed Algorithm

This chapter explains the phase-II of this project that also forms the proposed algorithm.

3.1 Image Segmentation

Segmentation of an image forms the first step of the algorithm prior to feature extraction and retrieval and has been carried out using JSEG. This assists the overall algorithm in deducing the properties (features) of each constituent region in an image. The essential idea of this segmentation technique is to separate the segmentation process into two independently processed stages, color quantization and spatial segmentation. In the first stage, colors in the image are quantized to several representing classes that can be used to differentiate regions in the image. This quantization is performed in the color space alone without considering the spatial distributions. Afterwards, the color information in the pixels is replaced by their corresponding color class labels, thus forming a class-map of the image. The main focus of this work is on spatial segmentation, where a criterion for "good" segmentation using the class-map is proposed. Applying the criterion to local windows in the class-map results in the "J-image", in which high and low values correspond to possible boundaries and interiors of color-texture regions. A region growing method is then used to segment the image based on the multi-scale J-images.

The results produced from the above segmentation method have all the disconnected regions classified as different regions. This becomes a cause of over-classification because often, spatially close but disconnected regions which get distinctly classified, are constituents of a single object or region which erraneously gets segmented (due to shadow, etc) into a number of parts. In order to correct these errors, a method is proposed to bring such separated regions under one region.

Consider any two segmented regions of the image. These regions are merged only if they satisfy the following conditions simultaneously,

1.

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 \leq \beta \tag{3.1}$$

Here, (x_1, y_1) and (x_2, y_2) define the centroids of the 'bounding boxes' for the two regions. A bounding box may be defined as the smallest possible rectangle that fully covers a region. β is the threshold value. Experiments have revealed the algorithm works optimally for $\beta = 10\sqrt{S_I}$, where S_I represents the image size in pixels.

2.
$$(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2 \leq \delta \tag{3.2}$$

Here, (R_1, G_1, B_1) and (R_2, G_2, B_2) represent the average RGB colour component values of the regions under consideration normalized to $[0, 1]$. δ is the threshold. The value of δ takes three different values depending upon the standard deviation existing in the average RGB values for all the regions in the image under consideration. The value is set to $\frac{10^{-4}}{3}$ if the standard deviation is less than 0.0654, to 10^{-2} if the standard deviation is greater than 0.083. Else, the value is kept 10^{-3}.

3.2 Image Annotation

Real Images are often composed of a wide variety of objects or regions. On many occassions, objects with similar texture, colour and other low level features belong to altogether different classes e.g. buildings made of stone and stone found on beaches or shores, in spite of their colour and texture being very similar, belong to different classes of objects. Thus, low level features such as colour composition, texture, etc alone are inadequate to handle the task of region identification.

Thus, it becomes imperative to first identify the 'representative or global-class' of the query image by using high level semantic information. Before defining the 'global class', it is required to define another term, 'context of an image' [18]. The wide variety of objects existing in a real image often hold a relationship amongst each other. It is this relationship among various regions that form the context of the image. Consider an example of beach. Consider a region existing in the image which has characteristics similar to stone. But stone can be in the form of stone-building or just stones on the beach. Upon introducting the concept of context, one can say from the general trend existing in images, that since that image has other regions as sky, sand, etc (beach), it is quite legitimate to take the ambiguous region as the stone on a beach rather than a stone-building. Thus, 'context' of an image is an important factor in order to identify the scenes and hence, the objects it contains.

The global-class can now be understood in terms of the context (the objects or regions and their relationships) of the Fig. 5.2a. The image contains snow mountains, spruce trees, sky and clouds. Thus, the global class of the image may be defined as the 'scenery'.

In view of above, the current system utilizes a system of image annotation to probabilistically identify the broad class of the image by segregating and annotating the regions existing therein (using ML based training algorithm with GMM distribution) and subsequently utilizing the annotations to list out similarly annotated images from the database. This whole procedure forms the first part of the overall algorithm.

We first describe the training methodology employing ML over GMM vector space. This is followed by the various features that have been used for the purpose.

3.2.1 Training using maximum likelihood (ML) over feature vector with Gaussian Mixture Model (GMM) distribution

Maximum likelihood is a common technique used for estimation of parameters defining a class. This method has a number of attractive attributes. First, it nearly always has good convergence properties as the number of training samples increases. Further, maximum likelihood estimation often can be simpler than alternate methods, such as Bayesian estimation techniques. Broadly, the technique attempts at maximizing a likelihood function, L. This function is simply the product of probabilities training samples belonging to a particular class under consideration. Mathematically, this is achieved by differentiating L with respect to the various parameters that need to be calculated and equating each of the parametric equation to zero.

The sample data in question can assume one of the many distribution models available. In the current work, the distribution model has been taken to be Gaussian that is given as follows,

$$f_X(\boldsymbol{x}/\theta_i) = \frac{1}{\sqrt{((2\pi)^k|\Sigma_i|)}} e^{-(\boldsymbol{x}-\boldsymbol{\mu_i})^T {\Sigma_i}^{-1} (\boldsymbol{x}-\boldsymbol{\mu_i})} \tag{3.3}$$

where, $\boldsymbol{x}$ is the feature vector for each region and $(\boldsymbol{\mu_i}, \Sigma_i)$ define the conditional distribution of the i^{th} class represented by θ_i.

The ML algorithm runs with an a-priori information about the number of clusters in a region. The current algorithm employs an automated methodology for calculating the number of clusters in a region. This is based upon the idea that the sum of probabilities for a feature vector belonging to a class should be maximum. This sum of probablities is calculated for each set of a certain number of cluster centers and the number for which this sum is maximum, is set as the number of clusters for that region. The graph presented in Fig. 1 shows the sum of probabilities for different number of cluster numbers. The red marker shows the number of clusters (eight in this case) for which the sum is maximum.

On following the above procedure, the following estimated of $\boldsymbol{\mu_j}, \Sigma_j$ are reached,

$$\boldsymbol{\mu_j} = \sum_{i=1}^{N} \boldsymbol{x_i} \tag{3.4}$$

and

$$\Sigma_j = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T \tag{3.5}$$

A single Gaussian estimate will not be sufficient to represent a region that might form multiple clusters in feature space. Thus, the current system employs GMM to generalize the classfication and reduce both training and test errors.

Once the training of the algorithm is done, it is then used to automatically annotate, as described in Chapter 3.2.3, all the images (including the training images) and generate the ACPV for each of them.

3.2.2 Feature Vector

The training algorithm mentioned in Chapter 3.2.1 employs a number of features constituting the feature vector. These features fall in two major categories to encode color and texture information.

3.2.2.1 LUV Colour Space

The LUV colour model has the advantage of being closer to human visual perception (over RGB). Since it is known to be perceptually uniform, a distance norm (Euclidian in this case) can be used to encode the non-linear response of the eye [8]. The motivation behind employing this colourspace instead of RGB or HIS is that it depicts a higher similarity between the norm computer distance and visual observation.

The features belonging to this group derive their values from the LUV colour model used for the image. There are a total of four features in the form of four central moments corresponding to each of L, U and V components. The moments are given as

$$M_l = \sum (l_i - \mu)^\gamma f_L(l_i) \tag{3.6}$$

$$M_u = \sum (u_i - \mu)^\gamma f_U(u_i) \tag{3.7}$$

$$M_v = \sum (v_i - \mu)^\gamma f_V(v_i) \tag{3.8}$$

Here, γ represents the order of the moment such that $\gamma \in \{1,2,3,4\}$. l_i, u_i and v_i represts the component values for the i^{th} pixel belonging to the region under consideration.

3.2.2.2 Texture

The set of features belonging to this group derive their values from the textural patterns existing within a region. The features are calculated by treating each region with sixteen different implementations (four scales and four orientation) of Gabor Filter. The rationale behind employing such a setting is to extract out the possible texture patterns that can exist within a region. Thus, this group accounts for a total of sixteen features. The Gabor Filter can be represented as follows,

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) e^{-\frac{1}{2}\left[\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+j2\pi(u_0x+v_0y)\right]} \tag{3.9}$$

Now a Gabor Filter with a scale m and orientation defined by n is represented as

$$g_{mn}(x,y) = a^{-m} g(x',y') \tag{3.10}$$

where,

$$x' = a^{-m}(xcos(\theta) + ysin(\theta)) \tag{3.11}$$

$$y' = a^{-m}(ycos(\theta) - xsin(\theta)) \tag{3.12}$$

$$\theta = \frac{n\pi}{K} \tag{3.13}$$

Here, K defines the total number of orientations (four in the present case). The feature set consists of mean and variance of the filtered image obtained from different filter implementations. The filter coefficients have been estimated using the method mentioned in [14]. Since gabor filter is originally designed for rectanglar regions, while the segmented regions can take arbitary shapes, the convolution is performed on a rectangle of maximum size over a region.

3.2.3 Annotation

Once an image is given to the system for retrieval, it is first segmented using the JSEG algorithm as discussed in Chapter 3.1. This is followed by the process of image annotation.

Firstly, the system estimates the posterior probabilities for each of the classes (seventeen in the current system). It has been determined empirically that out of the seventeen possible classes, the region has a plausibility of belonging to one of the classes corresponding to three highest probabilities. Hence, only the regions with top 3 probabilities are considered for further processing.

Once these three classes are selected for the region under consideration, they are checked for mutual region consistencies (MRC). The consistency for two regions can defined as the probability of three regions co-existing with one another. In case this frequency is low (inconsistent), the out of place region is eliminated, thus preserving the related classes. An example of 'inconsistency' would be the presence of shadow, white mountain and bus at the same time. It is quite obvious from both visual inspection and co-existance (as per the database) that white mountain and bus cannot appear together, while bus and shadow appear together with a high frequency. Thus the region has a much higher probability of being classified into bus or shadow and much less a white mountain. Hence, white mountain is removed from consideration. In the present system, the rules for MRC has been constructed based region co-occurances and visual inspection. An example of a similar structure can be found in [26].

Comprehensive experimentation reveals that for a region, if the probability difference between the top and any of the subsequent class candidates in the reduced result from above is greater than a certain factor, ϵ, then the region's likelihood of belonging to latter class(es) is almost nil. Hence, only the candidates lying within a range of ϵ of the highest probability are taken. This algorithm has been found to perform best with $\epsilon = 0.005$.

In case the number of possible classes for the considered region still remains 3, the third class is eliminated by taking the top 2 regions based on probability.

3.2.4 Intermediate Result Generation

The procedure till Chapter 3.2.3 yields the annotation wherein a few regions may be associated with more than one category. Before elaborating on the process of estimating similarly based on annotations, a annotation class probability vector (ACPV) for an image quantifying the presence of different classes needs to be calculated. This is done in the following manner. If the number of regions in the image are R then

1. the labels of the regions which have just one label are given a probability of $\frac{1}{R}$.
2. the multi-labelled classes have their labels alloted a probability of $\frac{1}{4R^2}$.

Following this, ACPV is created with each of its element corresponding to each of the classes. The value of each element is calculated as,

$$V_l = \sum_{i=1}^{17} P_{l,i} \tag{3.14}$$

where, V_l is the vector element and $P_{l,i}$ is the probability of the i^{th} region for belonging to the class l. Clearly, $\sum_l V_l = 1$.

Once the ACPV, V, is created it can be compared to the pre-calculated ACPVs of the database images. For comparison, Helinger Distance [6] is used which is defined as follows,

$$H(V_q, V_d) = \sqrt{1 - BC(V_q, V_d)} \tag{3.15}$$

where V_q and V_d are the ACPVs of the query image and the database image, respectively. $BC(V_q, V_d)$ is defined as the Bhattacharyya coefficient calculated as,

$$BC(V_q, V_d) = \sum_{i=1}^{N} \sqrt{v_q(i) v_d(i)} \tag{3.16}$$

Here, $v_q(i)$ and $v_d(i)$ define the i^{th} element of V_q and V_d, respectively and N is the number of classes.

Since the current database consists of 100 images belonging to each global class, a total of 200 images based upon the $H(V_q, V_d)$ have been taken as the intermediate result. This is done in order to ensure that, if not all, most of the images belonging to the correct global class are passed on the next step for further processing as expained below.

3.3 Region Based Image Retrieval

3.3.1 Features

The results obtained from the previous procedure based on annotation is further refined by employing RBIR. This retrieval is based on a number of low level features optimized for a region based search. The set of features and their corresponding distance metrics have been explained in the following sub-sections.

3.3.1.1 Texture Based Features

The texture based features are derived by using Gabor Filter already described in Chapter 3.2.2.2. Here, the Gabor Filter has been used with 6 scales with each being in 8 different orientations. Thus, a total of 48 different features are obtained from texture. Manhattan distance has been used to calculate the distance between the feature values so generated.

3.3.1.2 Colour Based Features

The colour based features desciptor used for extraction of colour information is MPEG-7 based dominant Colour descriptor [15]. This descriptor defines the set of dominant colour for a region and is defined as,

$$F = \{\{c_i, p_i, v_i\}, s\} \tag{3.17}$$

Here, c_i defines the i^{th} dominant colour, p_i defines its percentage value and v_i defines its variance. The spatial coherency s is a single number that represents the overall spatial homogeneity of the dominant colors in the image.

The dominant colours in a region are determined by employing a clustering approach (k-means in the present case). The two major issues with k-means i.e. convergence to local minima and initial number of cluster have been tackled. Since, a region is a particular segment of an image with uniformity in it's properties, the algorithm starts with an five initial cluster. The clusters are merged for two cases, particularly for empty clusters and when the distance between two clusters is less than a threshold, T_d. The dissimilarity between two feature sets is defined in [15] as follows:

$$D^2(F_q, F_d) = \sum_{i=1}^{N_1} p_{1i} +^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2\alpha_{1i,2j} p_{1i} p_{2j} \tag{3.18}$$

where, the subscripts q and d in all variables stand for descriptions F_q and F_d respectively, and $\alpha_{k,l}$ is the similarity coefficient between two colors c_k and c_l

$$\alpha_{k,l} = \begin{cases} 1 - \frac{d_{k,l}}{d_{max}}, & \text{if } d_{k,l} \leq T_d, \\ 0, & \text{if } d_{k,l} \geq T_d. \end{cases}$$

where, $d_{k,l} = ||c_k - c_l||$, T_d maximum distance for two colors to be considered similar and $d_{max} = \alpha T_d$. A normal value for T_d is between $10 - 20$ in the CIE-LUV color space and for α is between $1.0 - 1.5$.

In the current work, a variant of the above stated method is used. Firstly, the dissimilarity measure being used is defined as,

$$D^2(F_q, F_d) = 1 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2\alpha_{1i,2j} p_{1i} p_{2j} \tag{3.19}$$

The rationale behind this modification is that the first two terms of the original equation (percentage of regions) often exhibit random behaviour resulting in ambiguous values for $D^2(F_q, F_d)$ which can often lead to faulty segmentation results. Using one in place of the first two terms ensures that the dissimilarity measure has a higher dependency on the similarity coefficient $\alpha_{q,d}$, which considers the colour difference between regions. Such a strategy reduces the dependence of the feature (and its distance) on the segmentation process.

Further, as per the proposed method of employing a fixed T_d for constructing the feature (clusters) and measurement of distance, the current algorithm uses two different values for different occasions. During

the formation of the database the value of T_d is kept lower than usual, thus preventing disparate clusters in a region, emerging due to illumination effects, from merging. In case of over-merging, the feature will lose information and represent only the average values for a region. On the other hand, the value of T_d has been kept higher while retrieval of images. The assertion is that this will ensure a continuous value to the similarity measure, $\alpha_{q,d}$, for a greater range of distances, thus producing better results than the case where $\alpha_{q,d}$ abruptly becomes zero beyond a short range.

3.3.2 Final Result Generation

The final result for RBIR case is generated by taking a weighted linear sum of the individual distances estimated for each feature. The equation is defined as

$$d(i,j) = \sum_f W_f d_f(i,j) \tag{3.20}$$

where, W_f represents the weight assigned to the feature f. $d_f(i,j)$ is the normalized distance corresponding to feature f between region i of first (query) image and region j of the second (database) image. The distance $d_f(i,j)$ is calculated using the distance metric defined for feature f as given in Chapter 3.3.1.

The inital weights are based upon the best results. Although they are fixed for the present case, updations can be made by including a simple user-feedback mechanism [20].

The linear combination defined in the last step will work only when each of the individual distances lie in the range of 0 to 1. This is ensured by employing a Gaussian normalization startegy as done in [20].

$$d_f(i,j) = \frac{d_f(i,j) - \mu_i}{\sigma_i} \tag{3.21}$$

Here, (μ_i, σ_i) represents the mean and the variance of the all the distances calculated for the i^{th} feature.

This normalization has a better performance than linear normalization. Finally, the distance between the regions of the query and the database images is calculated using one-one region matching strategy. Hereby, a region in the query image is matched to the nearest region (least distance) in the database image.

$$D = \sum_i \min_j d(i,j) \tag{3.22}$$

In order to further match the results to human intuition, the distance contribution due to a region is weighted by the size of that region.

Chapter 4

Implementation

The current algorithm has been implemented on Matlab version 7.6. The DCPR [9] library was used for implementing the GMM based training algorithm. The software for JSEG segmentation algorithm was obtained from [13].

Chapter 5

Results and Discussions

To cover a wider range of applications in image retrieval, a database by WANG [25] containing images from different domains for benchmarking the algorithm has been used. The database contains 800 images, with 100 images from each of the 8 classes. These images differ from each other not only in distribution of colours, but also in number of objects in the foreground, textural patterns, size of dominant objects etc. On a whole, being a standard evaluation database, these images cover most of the cases occurring in real-time image retrieval applications.

Training of the class feature space using ML algorithm has been done by using thirty images belonging to each global class. It is to be noted that in order to test the robustness of the current annotation algorithm, the number of images used in the retrieval database is 800 although the number of training images have just been kept at 240.

The following portion of this section has been organized in the following manner. Firstly,ACPV for an image is demonstrated to show the distribution of probabilities among the seventeen classes. This is followed by explanation of the terms 'precision' and 'recall', most widely used to estimate the quantitative performance of a retrieval algorithms [24, 19, 16]. This is followed by depiction of the graph that is generated using the precision and recall values. After that, an explanation, using precision-recall graph, of the stark difference in the performance of annotation based method and RBIR in certain type of images is given. Following this, the performance of the CBIR system designed during the first phase of the project is compared with the current system of RBIR using precision-recall curve. This is followed by a detailed comparison of three techniques i.e. pure annotation based method, pure RBIR method and our algorithm, is done. Precision and recall concepts have been used to assess the performance of the three methods quantitatively. Lastly, a few more final results of the algorithm have been depicted.

Fig. 5.2b gives a pictorial representation of the ACPV created for the image shown in Fig. 5.2a.

The recall value of a result is defined with the following ratio

$$R = \frac{n_c}{N_T} \tag{5.1}$$

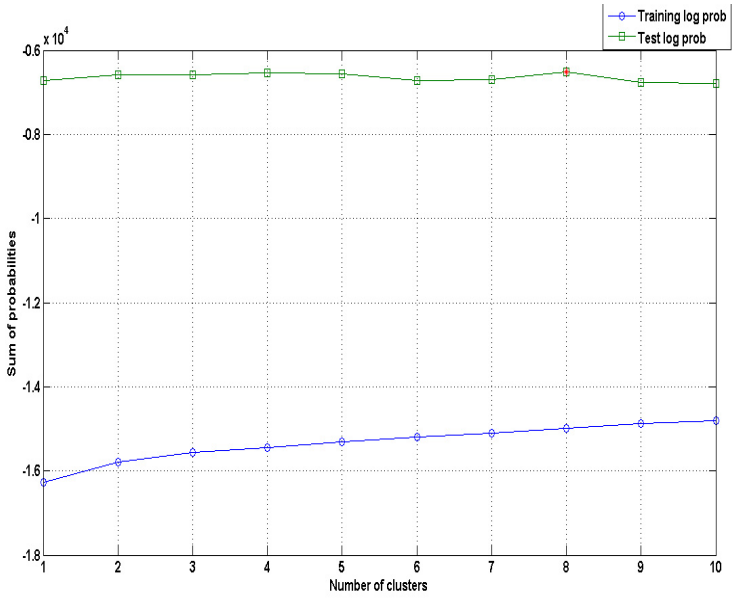
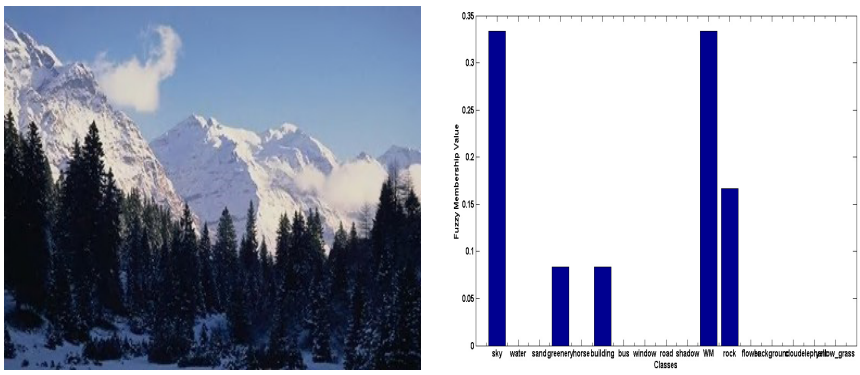


Figure 5.1 Graph showing the sum of probabilities value for different number of cluster centers



(a) (b)

Figure 5.2 (a) Image from class Mountain (b) ACPV for image in 5.2a

(a) (b)

Figure 5.3 (a)Image of a bus (b)Precision-recall curve for annotation based method and RBIR for image in 5.3a

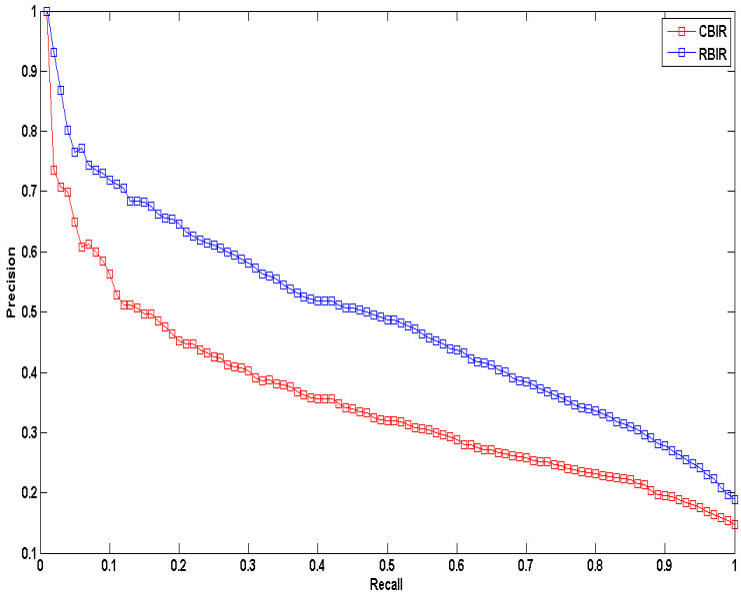


Figure 5.4 Average Precision-recall curves for CBIR (global features) and RBIR based retrieval algorithm

The precision value can be defined as,

$$P = \frac{n_c}{n_T} \tag{5.2}$$

Here, N_T is the total number of images in the database that belong to the same global class as the query image. n_T represents the total number of images produced in the retrieved result. n_c represents the total number of images in n_T that belong to the same global class as the query image in the retrieved result. For each such set, precision and recall values can be plotted to give a precision-recall curve. This curve has a distinctive saw-tooth shape. If the $(k+1)^{th}$ document retrieved is nonrelevant then recall is the same as for the top k documents, but precision has dropped. If it is relevant, then both precision and recall increase, and the curve jags up and to the right. Clearly, the higher the precision value for a given recall, the better is the performance of the algorithm for that recall.

In a ranked retrieval context, as in our case, appropriate sets of retrieved documents are naturally given by the top k retrieved documents. This can be explained in the following manner. The efficacy of web search results are usually estimated by assessing the good results on the first few pages. This leads to measuring precision at fixed low levels of retrieved results, such as 20 to 50 documents instead of the whole database [12]. This is referred to as "Precision at k".

Consider the graph shown in Fig. 5.3b for the image Fig. 5.3a. It clearly shows the difference in the results for pure annotation based retrieval and pure low level feature based retireval. This observation is quite common in cases where a single class may be existing in multiple forms i.e. red bus, blue bus, etc. The reason behind this is the difference of human perception, a high level feature based technique, to the low level feature extraction done by a computer. This can be explained with an example of retrieval of buses. In the training of annotation based algorithm, a bus will be labelled as a 'bus' irrespective of its colour. The algorithm will itself form different clusters for different coloured buses (GMM) with the class 'bus'. Thus, when the retrieval is done, the bus gets automatically annotated belonging to the class 'bus' irrespective of the colour. Thus, any image with a bus gets an annotation of 'bus' on one of its region and has a high likelihood to be matched with the bus images in the database as its colour will get matched to one of the clusters within the class bus. Now consider the case of low level feature based retrieval. Here, the colour of the bus, its size, etc play the role of markers for identifying it. Because the databse consists of images of flowers which may be of the same colour and relative size as that of the bus, there exist fair chances of their being listed among the top ranks for similarity with the bus in question. Thus, the results in this case would not be as good as for annotation based method.

Fig. 5.4 shows the comparison of the CBIR system based on global features and RBIR system based on the average precision-recall graph. The average precision is calculated by taking the average of the precision values for different query images for each recall level. Clearly, the CBIR system gives a lower performace than the RBIR system which in turn, as will be seen next, gives lower performance to the proposed system.

Now consider the graph shown in Fig. 5.5. The graph shows the performance of the three algorithms under test in terms of the precision-recall graph.

It is clearly evident from this graph the performance of the current algorithm is better than both, the pure annotation based method and pure RBIR. It should also be noted that the precision values

corresponding to the recall values from from 0.2 to 0.5 i.e. corresponding to $20-30$ relevant results, clearly shows particularly better performance for the current algorithm. This means that the algorithm is much better from practical point of view as most users look for the first 20-50 relevant results only.

Table 5.1 shows the numerical values of precision obtained for recall values with top 30, 40 and 50 images for the three algorithms. Here, precision@k represents the average precision for the test images for a retrieval of top k images. It is clearly seen that the values are higher for the current algorithm for all the recall values.

Table 5.1 Average Precision values observed at a retrieval of 30, 40 and 50 images

	Precision@30	Precision@40	Precision@50
Annotation	0.6103	0.5674	0.5145
RBIR	0.5533	0.5003	0.4661
our algorithm	0.6831	0.652	0.6081

Fig. 5.6, Fig. 5.7 and Fig. 5.8 shows the image results for the annotation based method, RBIR based method and current algorithm respectively. Looking at the result in terms of both, semantic class as well as low level features, the current algorithm gives better results than the other two.

For assessing the average performance of the current algorithm and comparing it with the other two algorithms, an average precision-recall graph for the algorithms has been shown in the Fig. 5.9. Here also, the current algorithm out-performs the other two.

A few more results of the current algorithm have been presented in Fig. 5.10, 5.11 and 5.12 for critical appraisal.

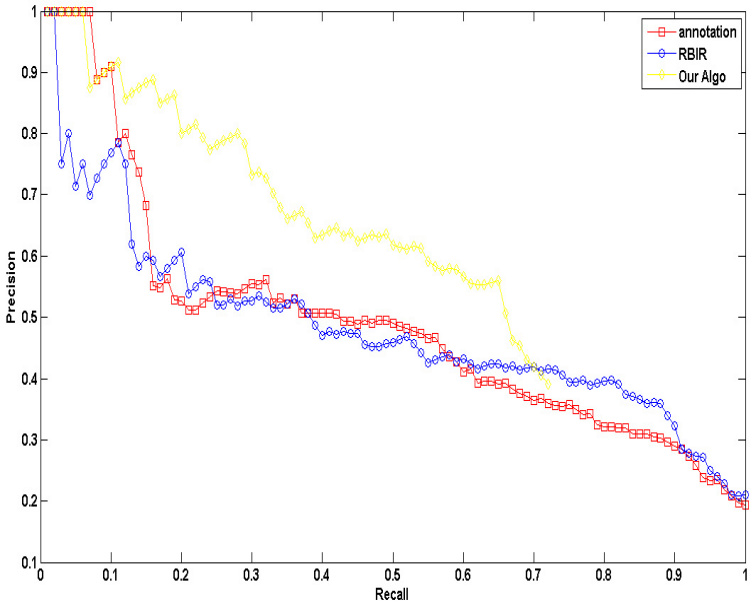


Figure 5.5 Precision-recall curve corresponding to three method for query image shown in Fig. 5.2a

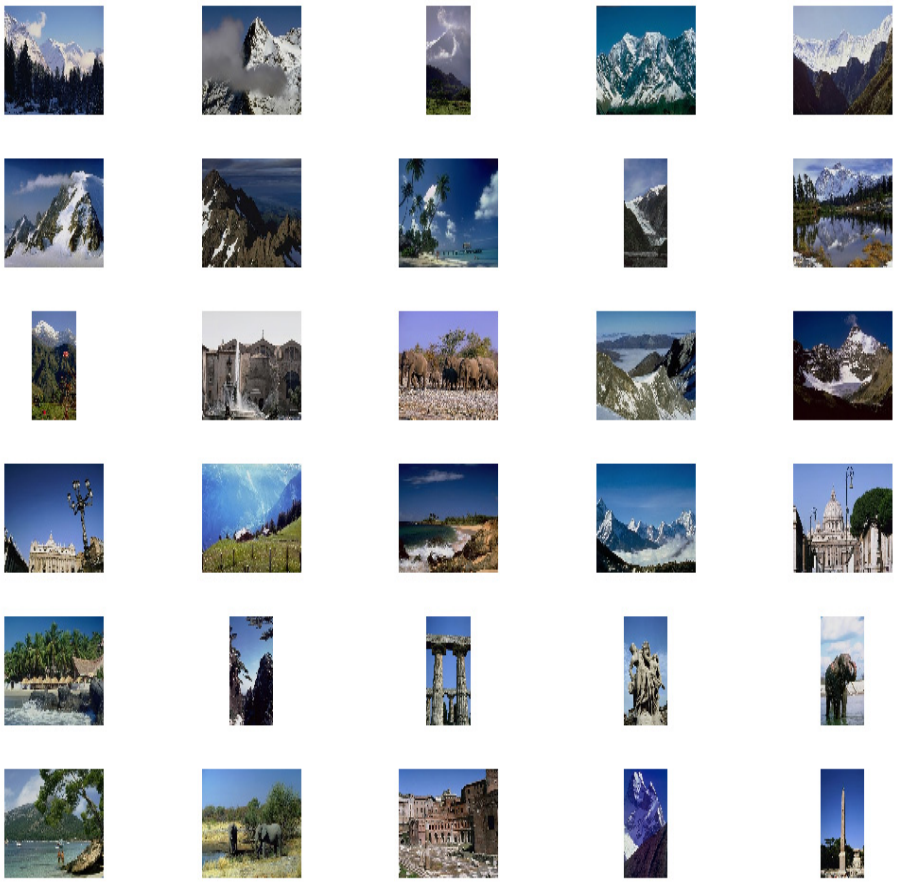


Figure 5.6 Result from annotation based retrieval (precision@30= 16)

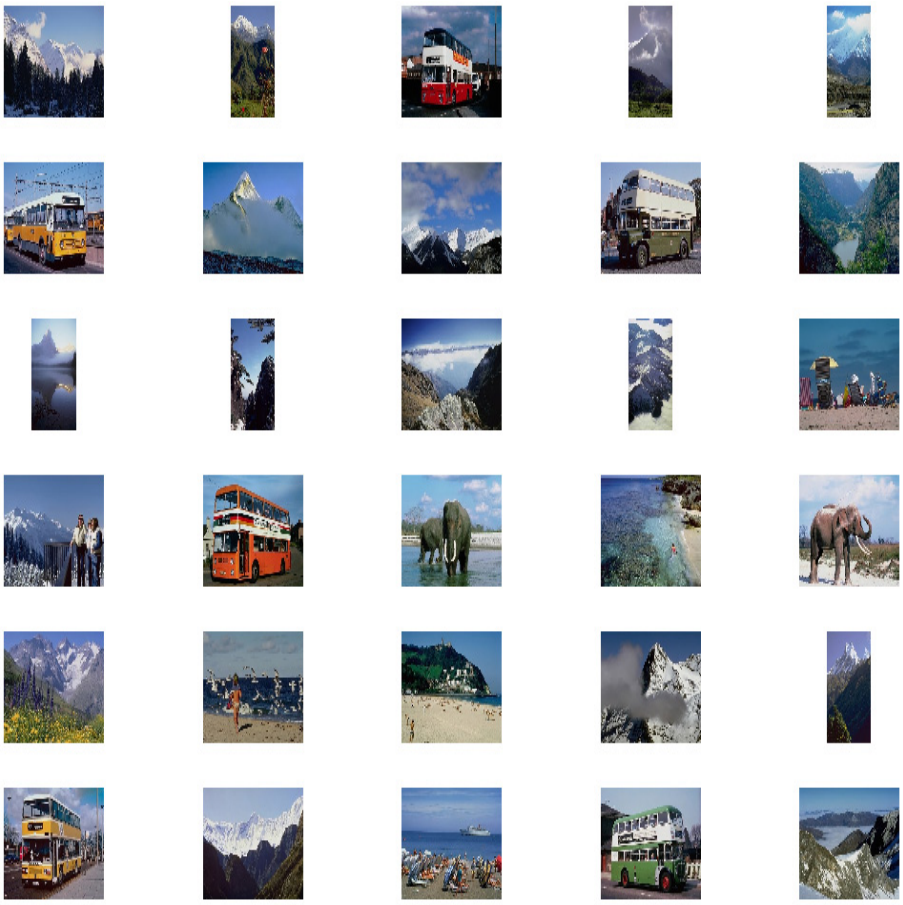


Figure 5.7 Result from RBIR (precision@30= 17)

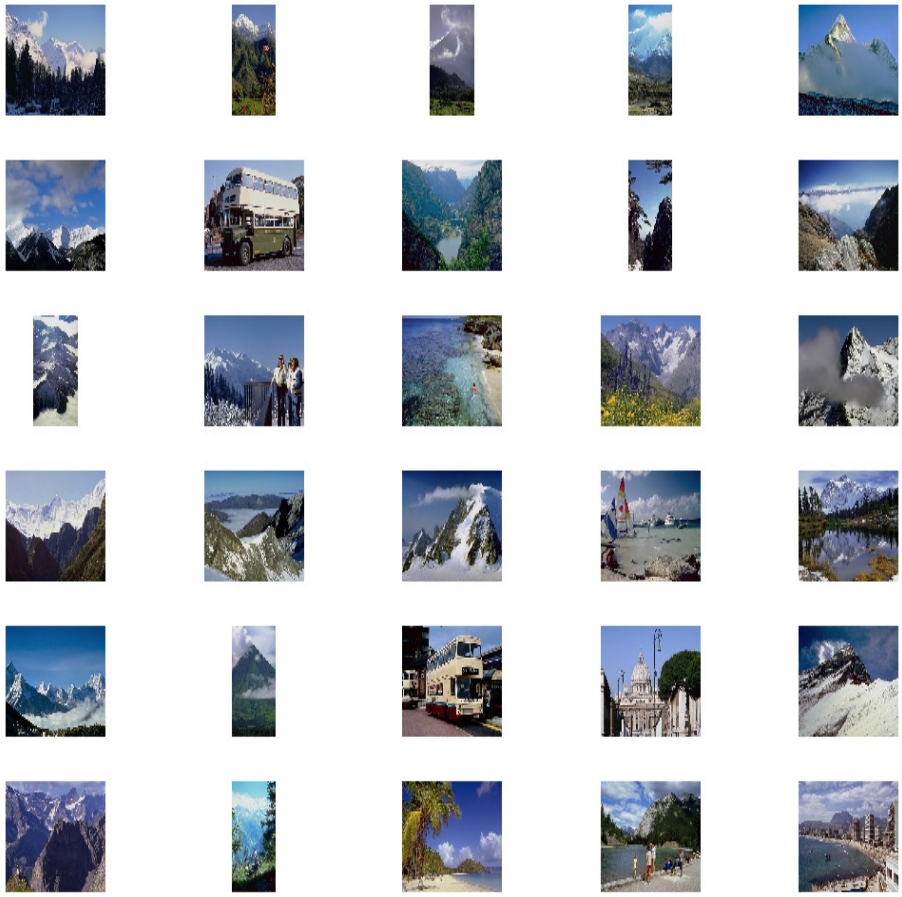


Figure 5.8 Result from our algorithm (precision@30= 23)

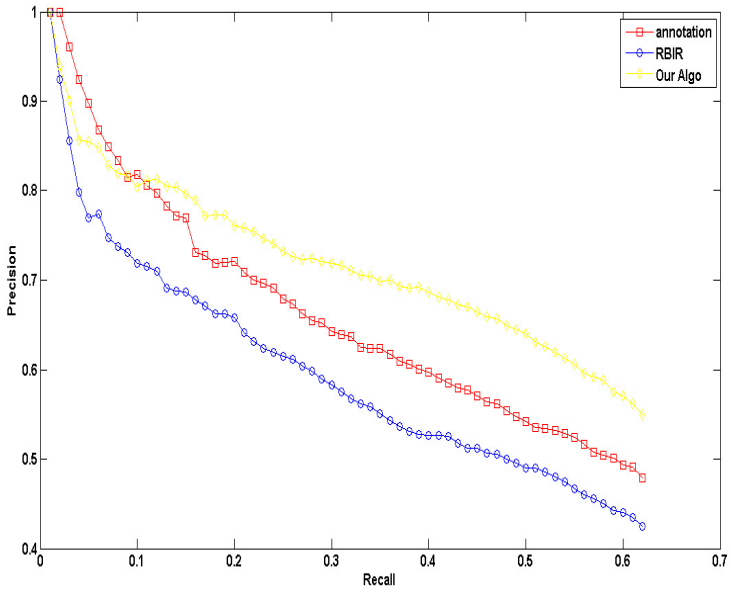


Figure 5.9 Average Precision-recall curve for the test images

Figure 5.10 Retrieval using our algorithm (precision@30= 30)

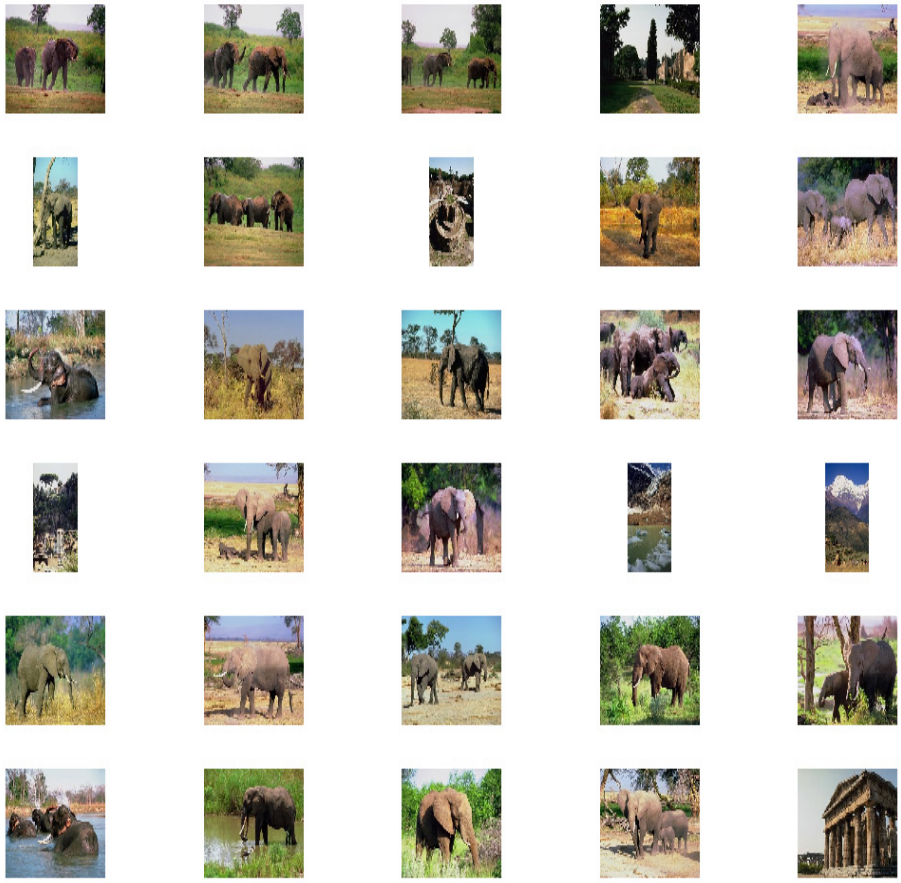


Figure 5.11 Retrieval using our algorithm (precision@30= 26)

Figure 5.12 Retrieval using our algorithm (precision@30= 30)

Chapter 6

Original contributions to the current work

The theory and implementation of annotation of regions in images for identification of the global class followed by the implementation of RBIR on the result so generated, is itself a new concept and does not exist in the literature. As has been explained in the report, the fundamental idea behind employing this methodology is to use image annotation first to identify the global class or the overall scene of the image and find images with similarly scene from the database. Once this is achieved, the selected images are compared to the query image using low level feature based region matching. This ensures a finer comparison of the image. There are several other novel ideas that have been implemented in and around the core techniques of image annotation and RBIR. These new contributions (in order of their occurrence in the algorithm) are listed below

6.1 Post segmentation merging

The results from JSEG have all the disconnected regions classified as different regions. This becomes undesirable in places where the same region gets disconnected due to obstructions or shadow. In order to remove the over classification, an innovative technique is proposed (Chapter 3.1) to bring such separated regions under one region. It uses the distance of the centroids of the two regions and their average colour values and checks it against certain thresholds to check for their similarity and possible merging.

6.2 Post fuzzy annotation label set reduction

The fuzzy annotation gives membership values to different labels for their being the correct one for a given region. This becomes a problem because often, images consist of as many as twenty five regions. Using seventeen membership values for each of this region sums up the total membership values for the whole image to four hundred and twenty five. This becomes a cumbersome computational problem. Moreover,

some of the membership values are too low to be of any significance and hence, become an unnecessary overhead. Inconsistent labels with respect to the context of the image also need to be removed. In order to deal with this situation a three-step procedure is formulated (Chapter 3.2.3). It chooses the three (or less) most likely labels for the region under consideration by looking the membership values and reduced this set by removing inconsistencies that might exist with regards to the context of the image. Finally, the region is left with a maximum of two labels, thus reducing the overhead by almost ten times.

6.3 Annotation class probability vector (ACPV)

This vector is designed to compare the images in the database to the query image based on the annotation (Chapter 3.2.4). This vector is calculated for the whole image. It consists of different values assigned to different labels in accordance to the likelihood of their occurrence in the image. Two images are compared by calculating Helinger Distance between their ACPVs.

6.4 Distance metric in Colour Descriptor

The contribution to this part mainly exist in the form of improvisation to the existing distance metric used for the comparison of two regions. The proposed dissimilarity measure is replaced by a new one which ensures for stability in the results and reduces the random behaviour resulting due to first two terms of the measure Further, as against the originally proposed method of employing a fixed threshold for constructing the feature (clusters) and measurement of distance, the current algorithm uses two different values for database creation (Chapter 3.3.1.2) and for comparison. This is done to serve the individual requirements for the two occasions and gives better performance.

Chapter 7

Future Work

Designing an image query system has been a rigrously researched area for quite sometime now. A number of new techniques and amalgamation of older ones are being proposed frequently. The current work belongs to the second class as an attempt has been made to use both, semantic information and low level features in synergy. The future work from here lies in trying various combination of techniques for catering to both, low level and high level features. For example, various low level features other than the ones proposed here can be used e.g. Tamura features, boundary extraction, etc. Neural Networks can be used for training of algorithm for semantic information. Another modification that can be done the introduction of multiple instance learning (MIL) [19] and thereby reducing the overhead of manual training. Further, since segmentation of the colour image is the starting step of the whole algorithm, it is imperative to do this work as accurately as possible. Although the current algorithm is performing more than satisfactorily, still there is scope of improvement in cases where the objects get merged (e.g. sand and elephant in the current database) or classified as different (e.g. due to the effect of shadow) erraneously.

Chapter 8

Conclusion

A new algorithm has been designed which successfully attempts at incorporating the high level semantic information along with low level features in order to effectively retrieve images based on an image query. The algorithm runs in two stages. The first stage tends to find the semantically similar subset of images from the database based on fuzzy annotations generated from a novel method. For this, an maximum likelihood based training algorithm has been used over feature vectors with Gaussian mixture model. This is followed by implementation of region based image retrieval algorithm on the so generated subset. For this, a colour descriptor is used to extract colour information and Gabor filtering is used with different scales and orientations to extract the texture information. The results have been assessed both qualitatively and quantitvely by using a graph of precision and recall values. A comparison is done between pure annotation based method and pure region based retrieval. The results clearly depict the supremacy of the current algorithm over these two methods. In addition, the RBIR system outperforms the CBIR system since a region based query is more closer to human intutition.

Bibliography

[1] Linda H. Armitage and Peter G.B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.

[2] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1026–1038, 2002.

[3] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001.

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

[5] R.M Haralick, K. Shanmugam Deng, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6:610–621, 2001.

[6] I. A. Ibragimov and R. Z. Khasminskii. *Statistical estimation : asymptotic theory / I. A. Ibragimov, R. Z. Hasminskii ; translated by Samuel Kotz.* Springer-Verlag, New York :, 1981.

[7] Qasim Iqbal and J. K. Aggarwal. Cires: A system for content-based retrieval in digital image libraries. In *in Invited Session on Content-based Image Retrieval: Techniques and Applications, 7 th International Conference on Control Automation, Robotics and Vision (ICARCV*, pages 205–210, 2002.

[8] Anil K. Jain. *Fundamentals of digital image processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[9] Jyh-Shing Roger Jang. Dcpr (data clustering and pattern recognition) toolbox. http://mirlab.org/jang.

[10] Thomas M. Lehmann, Mark O. Güld, Thomas Deselaers, Daniel Keysers, Henning Schubert, Klaus Spitzer, Hermann Ney, and Berthold B. Wein. Automatic categorization of medical images for content-based retrieval and data mining. *Computerized Medical Imaging and Graphics*, 29(2-3):143 – 155, 2005. Imaging Informatics.

[11] Y. Liu, D. Zhang, G. Lu, and W. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, January 2007.

[12] João Magalhães, Simon Overell, and Stefan Rüger. A semantic vector space for query by image example. In *in ACM SIGIR Conf. on research and development in information retrieval, Multimedia Information Retrieval Workshop*, 2007.

[13] B. S Manjunath. http://vision.ece.ucsb.edu/segmentation/jseg/software.

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

[15] B. S. Manjunath, Jens rainer Ohm, Vinod V. Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:703–715, 1998.

[16] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis. An ontology approach to object-based image retrieval. In *In Proc. IEEE Int. Conf. on Image Processing (ICIP03*, pages 511–514, 2003.

[17] H. Müller. A review of content-based image retrieval systems in medical applications clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, February 2004.

[18] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, December 2007.

[19] Nikhil Rasiwasia, Student Member, Pedro J. Moreno, and Nuno Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Trans. Multimedia*, 9:2007, 2007.

[20] Yong Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):644–655, 1998.

[21] Sven Siggelkow, Marc Schael, and Hans Burkhardt. Simba - search images by appearance. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 9–16, London, UK, 2001. Springer-Verlag.

[22] Yanfeng Sun, Hongjiang Zhang, Lei Zhang, and Mingjing Li. Myphotos: a system for home photo management and processing. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 81–82, New York, NY, USA, 2002. ACM.

[23] Christopher Town and David Sinclair. Content based image retrieval using semantic visual categories. Technical report, 2001.

[24] Julia Vogel and Bernt Schiele. Semantic modeling of natural scenes for content-based image retrieval. *Int. J. Comput. Vision*, 72(2):133–157, 2007.

[25] James Z. Wang. http://wang.ist.psu.edu.

[26] Gao Yan-Yu, Yin Yi-Xin, and Takashi Uozumi. A hierarchical image annotation method based on svm and semi-supervised em. *Acta Automatica Sinica*, 2008.