

Learning in CBIR

A Thesis submitted in the partial fulfillment
of the requirements for the award of degree of

Master of Technology

in

Computer Science

By

Ms. MounikaDhanraj



Department of Computer & Information Sciences

University of Hyderabad

Hyderabad , India

April, 2010

Certificate

This is to certify that the project work entitled “ **Learning in CBIR** ” being submitted by **Ms. MounikaDhanraj** (Reg.No: 08MCMT10), student of Department of CIS, school of MCIS, University of Hyderabad, in the partial fulfillment of the requirement for the award of degree of **Master of Technology in Computer Science** is a bonafide work carried out by her under our supervision.

Prof.B.L.Deekshatulu,
Project Supervisor,
Dept Of CIS,
University of Hyderabad.

Prof. Chakravarthy Bhagvati
Project Supervisor,
Dept Of CIS,
University of Hyderabad.

Prof.C.R.Rao
Project Supervisor,
Dept Of CIS,
University of Hyderabad.

Prof. Arun Agarwal
Head of Department,
Department of CIS,
University of Hyderabad

Prof. T. Amarnath
Dean,
School of MCIS,
University of Hyderabad

Acknowledgment

It gives a great pleasure to express my sincere thanks to my guide **Prof.B.L.Deekshatulu** for giving an opportunity to work with him. His profound knowledge and personal care have provided me with necessary fillip to make this project successful.

I am also extremely thankful to **Prof.Chakravarthy Bhagavathi** and **Prof.C.R.Rao** for their involvement and valuable guidance during the project discussions which helped me a lot to complete the project.

I express my sincere thanks to the head of department **Prof. Arun Agarwal**, for providing necessary computing facilities. I would like to thank to Dean **Prof. T.Amarnath**, School of Mathematics, Computers and Information Sciences for his valuable cooperation.I convey my heartfelt thanks to AI Lab Staff for their help in completing the project work successfully.

I take this as an opportunity to thank all my friends ,for their encouragement and cooperation completing this project successfully.

A final and heart full thanks to my parents.

With sincere regards
MounikaDhanraj

Abstract

Content Based Image retrieval is a task of retrieving images which are visually similar to the query image, from the database. Relevane feedback is commonly used to improve the performance of the CBIR system by incorporating user feedback iteratively. Main aim of our work is to infer a semantic space from the users relevance feedback so that the system will improve the performance of retrieval. This is accomplished by using both short term and long term learning strategies. In short term learning through relevane feedback the query vector is modified according to the relevant images feature vectors. A longterm learning creates a semantic space from the feedbacks given by the user in short term learning. The idea is that after several rounds of feedback the user has a pool of images relevant to the particular query, aggregating these results we construct a semantic space with a concomitant improvement in the system performance . We use SVD to reduce the dimensionality of semantic space both for savings in storage and for improvement in retrieval performance.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Definition	2
1.3	Organization of the Thesis	2
2	Outline of CBIR	3
2.1	Introduction	3
2.2	What is Content Based Image Retrieval?	3
2.3	Computational Features for CBIR Systems	4
2.3.1	Color	4
2.3.2	Texture	5
2.3.3	Shape Retrieval	5
2.3.4	Semantics	6
2.4	Problems with CBIR system	6
2.5	CBIR Systems and Applications	6
3	Literature Survey	9
3.1	Relevance Feedback	9
3.2	Relevance Feedback Techniques	10
3.2.1	Query Vector Modification	10
3.2.2	Short term learning and long term learning in CBIR:	12

3.3	Latent Semantic Indexing	13
3.3.1	Singular Value Decomposition	14
3.3.2	Mathematics underlying the SVD	14
3.3.3	LSI for Image Retrieval Systems	15
3.3.4	LSI Algorithm	15
3.3.5	Image search by LSI	16
3.3.6	Image Annotation with Relevance feedback and Latent Semantic Analysis	16
4	Approach	18
4.1	Present Procedure	19
4.2	Modified Procedure:	22
5	Experiments and Results	24
5.1	Discussion	24
6	Conclusions and Future Work	30
6.1	Conclusions	30
6.2	Future Work	31

List of Tables

5.1	Short term learning output	26
5.2	Long term learning output	27
5.3	Modified Long term learning output	27
5.4	First retrieval of the new query	28
5.5	Long term learning output of new query	28
5.6	Modified Long term learning output of new query	29

List of Figures

3.1	Illustration of QVM approach.a)The closest neighbors to the current queries are retrieved.b)A query reformulation can move the query to a region involving more relevant images.	11
4.1	Block diagram of our system	18
5.1	Query image	25
5.2	First Retrieval	25
5.3	First Retrieval	26
5.4	Queryimage.	27

Chapter 1

Introduction

Due to the rapid growth of number of digital images,there is an increasing demand for effective image management tools. Conventional content based image retrieval systems(CBIR) systems uses low level features automatically extracted from the images themselves to search for images relevant to a user's query. While there are research efforts to improve the performance by using different low-level features,and by modifying the similarity measures constructing from them,it is argued that,as unconstrained object recognition is still beyond the reach of current technology,these content based systems can at best capture only pre -attentive similarity ,not the semantic similarity.

1.1 Motivation

In regular relevance feedback schemes a CBIR system first returns a set of initial results to users based on query samples then it requests the users to mark it relevance on results . After incorporating users feedback the CBIR will learn the users feedback and return the refined results .These feedback procedures typically need to be iterated many times.

Our motivation is to use the users feedback history to capture the semantics of the images ,and infer the semantic space from the user's feedback so that the sys-

tem will improve the retrieval performance through accumulated user interactions. This is accomplished by using both the short term learning and long term learning strategies, where the long term learning is responsible for constructing the semantic space.

1.2 Problem Definition

The fundamental idea of this method is to address the limitations of the current systems with regard to searching for images at the semantic level. A long term learning method that creates a semantic space implicitly based on user's interaction in a relevance feedback and a short term learning are integrated to improve the performance of the retrieval. The idea is that after several rounds of relevance feedback the user has a pool of images relevant to his query, aggregating these results we construct a semantic space, with a concomitant improvement in the system performance. We use the Singular Value Decomposition (SVD) to reduce the dimensionality of the semantic space and for the improvement in the retrieval performance. SVD helps to correlate the relevance feedback's from different search sessions and reduce the subjectivity and noise introduced by the individual users. Due to the dimensionality reduction, the relevant and non relevant images are no longer linearly separable in this case a linear discriminant function is used to learn the target function for retrieving the relevant images.

1.3 Organization of the Thesis

The report is organized as follows. Chapter 2 gives an overview of traditional CBIR systems. Chapter 3 is about the literature survey. Algorithm is explained in chapter 4. Experimental results are given in chapter 5. Conclusion and future work in chapter 6.

Chapter 2

Outline of CBIR

2.1 Introduction

Digital contents are being generated at a dramatically speed. As the amount of collections of digital images increase, the problem of locating a desired image in an enormous collection also becomes very difficult. Therefore the need of an efficient method to retrieve digital images is recognized by the public. There are two approaches to image retrieval, Text Based approach and Content Based approach. The former solution is a more traditional approach which indexes images by using keywords. The keyword indexing of digital images is useful but requires a considerable level of effort and often limited for describing image content. The alternate approach, the Content-Based image retrieval indexes images by using the low level features of the digital images and the searching depends on features being automatically extracted from the image.

2.2 What is Content Based Image Retrieval?

Content Based Image Retrieval is the term used to describe the process of retrieving images from a database on the basis of the internal features of images. In CBIR, digital images are indexed by summarizing their visual contents through automatically ex-

tracted features such as texture,color and shape.There exist different ways to express the query.The query can be defined by submitting one or more examples images,providing a rough sketch of the desired item or by providing textual descriptions of the object.

CBIR retrieves stored digital images from a collection by comparing features extracted from the images.The most common features used are mathematical measures of color,texture or shape.The CBIR system [1],[2],[3] identifies those stored images whose feature values match those of the query most closely,and displays these found images to the user.In the following section some of the frequently used types of features used for image retrieval will be described.

2.3 Computational Features for CBIR Systems

2.3.1 Color

Color has been the most effective feature and almost all systems employ colors. Although most of the images are in the RGB (Red, Green, Blue) color space, this spaces only rarely used for indexing and querying as it does not correspond well to the human color perception. It only seems reasonable to be used for images taken under exactly the same conditions each time such as trademark images. Other spaces such as HSV (Hue, Saturation, Value) or the CIE Lab and Luv spaces are much better with respect to human perception and are more frequently used. This means that differences in the color space are similar to the differences between colors that humans perceive. The CIE, which stands for International Commission on Illumination,defined three standard primaries (X, Y, and Z) to replace red, green, and blue, because all visible colors could not be specified with positive values of red, green and blue components.(In other words, not all visible colors can be specified in the RGB color space! at least not with positive coefficients.) With this newly created

X,Y, and Z primaries, all visible colors could be specified with only positive values of the primaries. The CIE LUV color space is a perpetually uniform derivation of this standard CIE XYZ space. The transformation of RGB space to CIE L*u*v* or CIE L*a*b* space can be found in [19].

2.3.2 Texture

Texture is another important property of images. Texture features of images refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity. An images texture content provides information of image properties such as smoothness, coarseness, and regularity which is useful in a CBIR system. Various texture representations have been investigated in pattern recognition and computer vision. Basically, texture representation methods can be classified into two categories: structural and statistical. Structural methods, including morphological operator and adjacency graph, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular. Statistical methods, including Fourier power spectra, co-occurrence matrices, shift-invariant principal component analysis (SPCA), Tamura feature, World decomposition, Markov random field, fractal model, and multi-resolution filtering techniques such as Gabor and wavelet transform, characterize texture by the statistical distribution of the image intensity.

2.3.3 Shape Retrieval

Shape features of objects or regions have been used in many content-based image retrieval systems. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. Since robust and accurate image segmentation is difficult to achieve, the use of shape features for image retrieval has been limited to special applications where objects or regions are readily available. The state-of-art methods for shape description can be categorized

into either boundary or region-based methods. A good shape representation feature for an object should be invariant to translation, rotation and scaling .

2.3.4 Semantics

Most current CBIR systems retrieve images from a collection on the basis of the low level features of images,such as color,texture and shape.Nevertheless,some systems attempt to find images that are semantically similar to a given query.Semantically similar is meant in the sense of human visual similarity perception(or called high level in CBIR).

2.4 Problems with CBIR system

The performance of CBIR systems is not satisfied due to the following reasons.

- *The gap between high level concepts and low level features.* The assumption that the CBIR approach makes is that the high level concepts to low-level features mapping is easy for the user to do.While,in some cases the assumption is true,e.g.,mapping a high-level concept to low level features,in other cases,this may not be true.One example is to map an ancient vase with sophisticated design to an equivalent representation using low-level features.A gap exists between the two levels
- *The subjectivity of human perception.*Different persons,or the same person under different circumstances,may perceive the same visual content differently.This is called as human perception subjectivity.The subjectivity exists at various levels.

2.5 CBIR Systems and Applications

- **Commercial CBIR systems**

1. QBIC:<http://www.qbic.almaden.ibm.com/>.
2. Virage:<http://www.virage.com/online/>.
3. Excalibur:<http://www.excalib.com/>

- **Experimental Systems:**

1. Photobook
2. Chabot
3. Visuals Eek
4. MARS
5. Informedia
6. Surf image
7. Netra
8. Synapse

- **Practical applications of CBIR:** A wide range of possible applications for CBIR technology has been identified. Potentially fruitful areas include:

1. Fashion and Interior Design Similarities can also be observed in the design process in other fields, including fashion and interior design. Here again, the designer has to work within externally imposed constraints, such as choice of materials. The ability to search a collection of fabrics to find a particular combination of color or texture is increasingly being recognized as a useful aid to the design process.
2. Crime Prevention Law enforcement agencies typically maintain large archives of visual evidence, including past suspects facial photographs, fingerprints, Tyre treads and shoe prints. Whenever a serious crime is committed, they can compare evidence from the scene of the crime for its similarity to records in their archives.

3. Intellectual Property Trademark image registration, where a new candidate mark is compared with existing marks to ensure that there is no risk of confusion, has long been recognized as a prime application area for CBIR.
4. Medical Diagnosis While the prime requirement for medical imaging systems is to be able to display images relating to a named patient, there is increasing interest in the use of CBIR techniques to aid diagnosis by identifying similar past cases.
5. Architectural and engineering design.
6. The military.
7. Architectural and engineering design.
8. Journalism and advertising.
9. Medical diagnosis.
10. Geographical information and remote sensing applications.
11. Cultural heritage.
12. Education and training.
13. Home entertainment.
14. Web searching.

Chapter 3

Literature Survey

3.1 Relevance Feedback

Relevance Feedback is commonly used to improve the performance of CBIR system by allowing incorporation of user relevance feedback iteratively. The main idea behind Relevance feedback is as follows: the user submitting the query image to the system, it compares the query image to each image in the database and retrieve the images that are nearest to the query image. User marks some of the images as relevant and non relevant according to his/her information needs. If the user is not satisfied with the results, system updates the feature vector using relevant and non-relevant images to get better results in next iterations. The process is repeated until the user is satisfied or the results cannot be further improved. The RF techniques provide a way to bridge the gap between the machine subject in terms of low-level features and the human subject that is driven by high-level semantics.

3.2 Relevance Feedback Techniques

Let the query image and a data base image be represented by feature vectors $X = (x_1, x_2, x_3, \dots, x_d)$ and $Y = (y_1, y_2, \dots, y_d)$ respectively, where d is the number of selected features and x_i and y_i are the values of the i th feature. The system derives the similarity between X and Y by computing the distance under the given dissimilarity metric. The normalized Euclidean metric

$$Dist(X, Y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2 / d} \quad (3.1)$$

is generally used for this purpose. The top t database images that are the nearest neighbors of the query are then returned to the user. If the user is not satisfied with the retrieved results, he or she can activate an iterative RF process until satisfied.

3.2.1 Query Vector Modification

The query vector modification (QVM) approach repeatedly reformulates the query vector through users feedback so as to move the query towards relevant images and away from non relevant ones. Let a user submit the i th database image as the query and have experienced j RF iterations and let x_i^j denote the current query formulation. Also, let the set of relevant images identified at the j th iteration be R , and the set of identified non relevant images be N . For the $j + 1$ th RF iteration, the reformulated query vector is calculated as

$$X_i^{j+1} = \alpha X_i^j + \beta \sum_{Y_k \in R} \frac{Y_k}{|R|} - \gamma \sum_{Y_k \in N} \frac{Y_k}{|N|} \quad (3.2)$$

where Y_k are images that belong to region R or N , and α, β, γ are the parameters controlling the relative contribution of each component. The effect of the QVM can be illustrated in fig 3.1. For simplicity let us assume that only two features f_1 and f_2 are used for image similarity matching. All of the database images spreading over the feature space are either relevant (indicated by symbol “+”) or non relevant (indicated

by symbol “-”) to the current query x_i^j according to the user’s intention. However the most similar to the query according to the machine subject(KNN algorithm) in the sense of Euclidean metric are those residing at the interior of the circle centered at(see Fig.3.1a) .Thus,a simple query reformulation derived from the QVM formula can move query to a location that is expected to be in a region involving more relevant images.

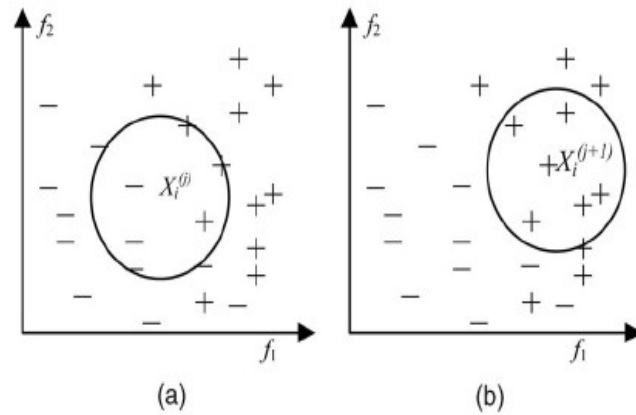


Figure 3.1: Illustration of QVM approach.a)The closest neighbors to the current queries are retrieved.b)A query reformulation can move the query to a region involving more relevant images.

3.2.2 Short term learning and long term learning in CBIR:

Short term learning uses the feedback information given by the user to improve the performance of the current relevance feedback session and long term occurs over the course of many query sessions. Short term learning should achieve improvements in retrieval in current query session ,where as long term learning should achieve improvements in retrieval performance of future query sessions.

One of the approach of CBIR uses both short term and long term learning .The system integrates both positive and negative feedback from all iterations to construct a semantic space to remember the users intent in terms of high level hidden semantic features. The short term learning refines the query by updating its associated weight vector using both positive and negative examples together with the long term learning based semantic space. The similarity score is computed as the dot product between the query weight vector and the high level features of each image stored in the semantic space.

Another approach to long term learning strategy using the historical retrieval information is presented. The approach presents adopts a semantic covering set constructed dynamically to deposit the user's historical retrieval information produced in previous process ,and predicts the semantic correlation between the images in the database and query sample according to the historical retrieval information when carrying out a new query session.

3.3 Latent Semantic Indexing

A fundamental deficiency of current information retrieval methods is that the words used often are not the same as those by which they seek the information has been indexed. There are two sides of the issue; we call them broadly *polysemy* and *synonymy*. Latent semantic indexing is an approach to automatic indexing and retrieval. It is designed to overcome the fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The problem is that users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user.

Latent semantic indexing analysis uses singular value decomposition. A large matrix of term-document association data is taken and a “*semantic*” space is constructed wherein terms and documents that are closely associated are placed near one another. Singular-value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that did not actually appear in a document may still end up close to the document, if that is consistent with the major patterns of association in the data. Position in the space then serves as the new kind of semantic indexing, and retrieval proceeds by using the terms in a query to identify a point in the space, and documents in its neighborhood are returned to the user.

3.3.1 Singular Value Decomposition

Singular value decomposition is closely related to a number of mathematical and statistical techniques in a wide variety of other fields, including eigen vector decomposition, spectral analysis, and factor analysis. We will use the terminology of factor analysis, since that approach has some precedence in the information retrieval literature. Latent semantic structure analysis starts with a matrix of terms by documents. This matrix is then analyzed by SVD to derive our particular latent semantic structure model.

For information retrieval purposes, SVD can be viewed as a technique for deriving a set of uncorrelated indexing variables or factors; each term and document is represented by its vector of factor values. Note that by virtue of the dimension reduction, it is possible for documents with somewhat different profiles of term usage to be mapped into the same vector of factor values. This is just the property we need to accomplish the improvement of unreliable data proposed earlier. Indeed, the SVD representation, by replacing individual terms with derived orthogonal factor values, can help to solve the fundamental problems.

3.3.2 Mathematics underlying the SVD

Any rectangular matrix $t \times d$ of terms and documents, X , can be decomposed into the product of three other matrices:

$$X = T_0 S_0 D_0'$$

(3.3)

such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. This is called the singular value decomposition of X . T_0 and D_0 are the matrices of left and right singular vectors and S_0 is the diagonal matrix of singular values. Singular value decomposition (SVD) is unique up to certain row, column and sign permutations and by convention the diagonal elements of S_0 are constructed to be all positive and ordered in decreasing magnitude.

The amount of dimension reduction i.e., the choice of k , is critical to our work. Ideally, we want a value of k that is large enough to fit all the real structure in data, but small enough so that we do not also fit the sampling error or the unimportant details. The proper way to make such choices is an open issue in the factor analytic literature. In practice, we currently use an operational criterion a value of k which yields good retrieval performance.

3.3.3 LSI for Image Retrieval Systems

Image can be viewed as vector of pixels; every pixel is described by its color. The vector of pixel represents some kind of keywords in image. But human observer extracts from image important features that define semantics of image for him. The man doesn't think about pixel but about persons or objects on image. So we need technique that is able to extract this features and that is resistant to minor changes of images (e.g. amount of light, contrast and moves of objects on the images). Direct usage of keyword based systems leads to results that are sensitive to small change of any keyword (pixel in query).

3.3.4 LSI Algorithm

1. Compute the co-ordinates of all images in the k -dimensional space by the partial SVD of the pixel-image matrix A

$$[U, S, V] = svds(A, k);$$

2. Compute the coordinates of the query vector q

$$qc = q' \times U \times inv(S);$$

3. Compute the similarity coefficients between the coordinates of the query vector and images.
 for $i = 1:n$ (Loop over all images)

4. $sim(i) = (qc \times V(i, :)' / (norm(qc) \times norm(V(i, :))));$

The following procedure returns to the user the vector of similarity coefficients sim . The i th element of the vector sim contains a value which indicates the measure of semantic similarity between the i th image and the query image. The increasing value of the similarity coefficient indicates the increasing similarity.

3.3.5 Image search by LSI

LSI method has been presented to achieve improvement in the image retrieval performance. The LSI method extracts different physical features from important objects of an image, and constructs a semantic space, each dimension of which represents a different feature component of the image. LSI discovers the latent relations among the images which are actually similar in semantic. The user feedback also been integrated into the LSI method so as to improve the feedback performance of the system.

3.3.6 Image Annotation with Relevance feedback and Latent Semantic Analysis

Automatic annotation of images is implemented on partially annotated databases by learning image concepts relationships from positive examples via inter-query learning. Latent semantic analysis is applied to an image/session matrix where relevance

feedback examples are collected from a large number of query sessions. SVD is exploited during LSA to propagate image annotations using only relevance feedback information.

These are some of the LSI based image retrieval techniques. In the next chapter usage of LSI and short term and longterm learning strategies are explained.

4.1 Present Procedure

1. Extract the low level features of all the database images and also the query image. The low-level features we considered here are color, entropy, contrast and occupancy ratio. Suppose the feature vector is of dimension n , and we have a database of m images then, a matrix of dimension $m \times n$ is constructed, where each row of a matrix corresponds to the feature vector of the database image and each column represents the features. Say this matrix as B .

In a similar manner extract the features of the query image, it will be vector of dimension $1 \times n$.

2. Compute the Manhattan distance between the database image feature vector and the query vector.
3. After computing the distance the top 20 images are displayed to the user to give the feedback. The images which are relevant to the query is marked as 1 and the images which are non-relevant are marked as 0. Thus we get a feedback vector for the query image which of size 20×1 . We call the feedback matrix as F .
4. By using the feedback matrix F , modify the query vector by taking the average of the features of all the relevant images to that query.
5. Compute the dot product between the modified query vector and the feature vector of the database image which was given feedback. Set a threshold θ and if the value of the dot product exceeds that threshold keep the corresponding value as one else keep the corresponding value as zero.
By this we will get a matrix H of dimension 20×1 . The matrix H is the machine generated feedback matrix.
6. Compare this machine generated matrix H with the feedback matrix F and update the query vector by a factor of α according to positive mistakes or

negative mistakes.

A mistake driven learning algorithm is used to update the query vector q . If the current classifier labels an image x as irrelevant while the user labels x as relevant we say a positive mistake occurs. If the current classifier labels x as irrelevant while the user labels x as relevant we say a negative occurs. The algorithm updates the weight vector as follows

- Negative mistake

$$q_i^{t+1} = \begin{cases} \frac{q_i^t}{\alpha} & \text{if } x_i = 1 \\ q_i^t & \text{if } x_i = 0 \end{cases}$$

- Positive mistake

$$q_i^{t+1} = \begin{cases} q_i^t, & \text{if } x_i = 0 \\ 1, & \text{if } x_i = 1 \text{ and } q_i^t = 0 \\ \alpha q_i^t, & \text{if } x_i = 1 \text{ and } q_i^t \neq 0 \end{cases}$$

7. Compute the dot product between the above modified query vector and the images in the database for which the feedback was given and again generate a new H . Repeat steps 5 and 6 until the H matrix is stabilized.
8. Using the stabilized H matrix and a classifier tune H to the whole database and get the H matrix of size $m \times 1$. Repeat steps 1 to 8 for all k queries to get the matrix H of size $m \times k$.

NOTE: Short term learning is always used only for that query image

9. The above steps end the short term learning session. For storage and performance requirements SVD is computed on H matrix. The rank of the reduced

matrix is equal to the number of classes of images in the database.

$$H = [U \times S \times V']$$

The B matrix which contains the low-level features is replaced by the product of $U \times S$. This will be considered as the semantic space.

10. When a new query other than the k queries, first extract the low level features and compute the Manhattan distance between original B and new query feature vector.
11. Get the feedback for top 20 images and mark relevant and non relevant images. Call this feedback matrix as F_n which consists of ones and zeroes. The images which is relevant to the query is given a one otherwise a zero.
12. Modify the query vector by taking the average features of the relevant images from the modified B.
13. Compute the dot product between the modified B and the modified new query vector to generate the matrix H. Compute the dot product between the modified query vector and the feature vector. Set a threshold and if the value of the dot product exceeds that threshold keep the corresponding value as one else keep the corresponding value as zero. By this we will get a matrix H which consists of ones and zeroes.
14. Compare the matrix H with the feedback matrix F and update the query vector by a factor of according to positive mistakes or negative mistakes.
15. Compute the dot product between the above modified query vector and the images in the database for which the feedback was given and again generate a new H.
16. Repeat the steps 14 and 15 until the H matrix is stabilized.

17. Thus the final results are obtained.

4.2 Modified Procedure:

1. Construct matrix B of size $m \times n$, matrix Q of size $1 \times n$ and matrix F of size $1 \times k$, where m corresponds to the number of images in the database and n corresponds to the number of features, K corresponds to the query images.
2. Compute the Manhattan distance between the matrix B and Q and get a resultant matrix of size $m \times 1$.
3. Rank order the values in the R matrix and display the top 20 images to the user.
4. From the displayed images the user marks relevant and non relevant images, apply Query Vector Modification(QVM) to modify the query vector .
5. Compute the dot product between the modified query vector and the B matrix and get resultant R matrix. Set a threshold and ,if the value in the R matrix exceeds a threshold keep a 1 otherwise keep a 0 in this manner we get a matrix H of size $m \times 1$.
6. Compare the H matrix with F matrix and modify the query vector by a vector of alpha according to the positive or negative mistake by using mistake driven learning algorithm.
7. Repeat the steps 5 and 6 until the hamming distance between the F and H matrices is minimized. Repeat the steps 1- 6 for k queries to get H of dimension $m \times k$.
8. This ends the short term learning session and after this appends the matrix H to the matrix B. By appending the H matrix to the B matrix the low-level

features are combined along with the relevant and non relevant information of an image to the particular query. Compute the svd of the matrix B augmented with H to reduce the space.

$$[U, S, V] = svd(B|H)$$

Now the B new will be the product of $U \times S$.

9. For a new query extract the low-level features and compute the distance between the new query feature vector and the B matrix which contains the database image feature vectors, display the top 20 images to the user to give the feedback.
10. From the feedback given by the user pick the relevant images feature vectors and non relevant feature vectors from the matrix B_n (it is a modified B matrix which is considered as the semantic space.) ,and apply the QVM to obtain the modified query feature vector.
11. Compute the dot product between the modified query vector and the B_n ,set a threshold and if the corresponding value exceeds that threshold keep a one other wise keep a zero.This matrix is considered as the H matrix.
12. Compare this H matrix with the feedback matrix and modify the query vector according to the mistake driven algorithm.
13. Repeat the above step until the hamming distance between the H and F is minimized.
14. Display the images to the user which are having there values as one in H matrix.Thus the final results are obtained.

Chapter 5

Experiments and Results

5.1 Discussion

Fig 5.1 is the query image. Given this query image first retrieval(fig5.2 and fig5.3) are the results which are obtained by computing the Manhattan distance between the query images and the database images. Table5.1 shows the short term learning output which is obtained by modifying the query vector by taking the average of the relevant image feature vectors and then updating it by a factor of alpha according to positive or negative mistakes as explained in the procedure(4.1).

Table 5.2 and 5.3 shows the long term learning outputs. Here the performance of these methods are same with respect to the query which is given in figure5.1. Table 5.5 and table 5.4 shows the long term learning outputs of the new query which is not there in the database.

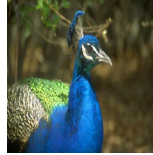


Figure 5.1: Query image



Figure 5.2: First Retrieval

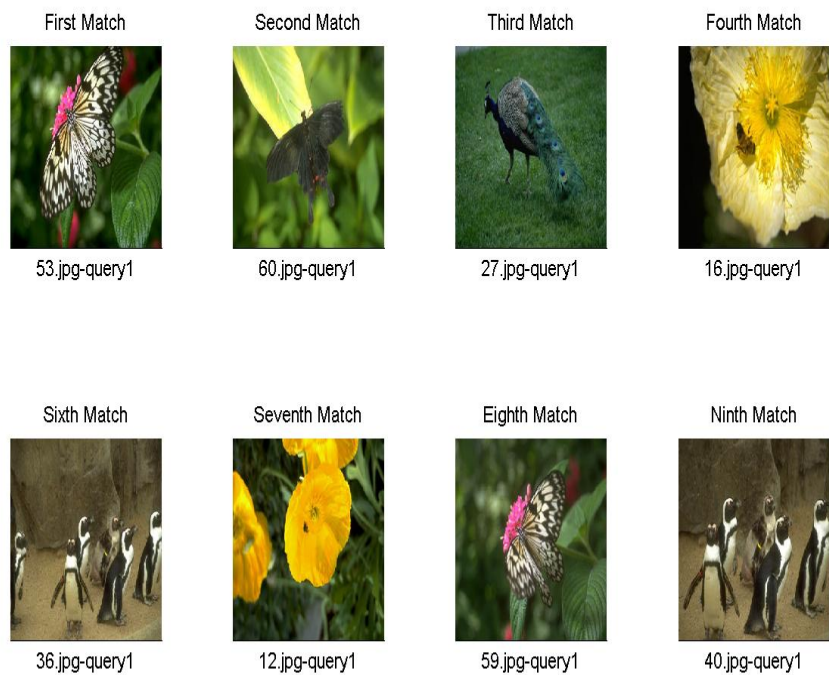


Figure 5.3: First Retrieval

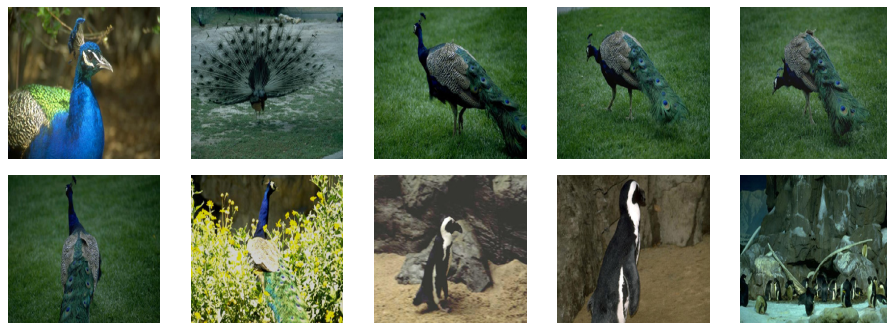


Table 5.1: Short term learning output



Table 5.2: Long term learning output



Table 5.3: Modified Long term learning output



Figure 5.4: Queryimage.

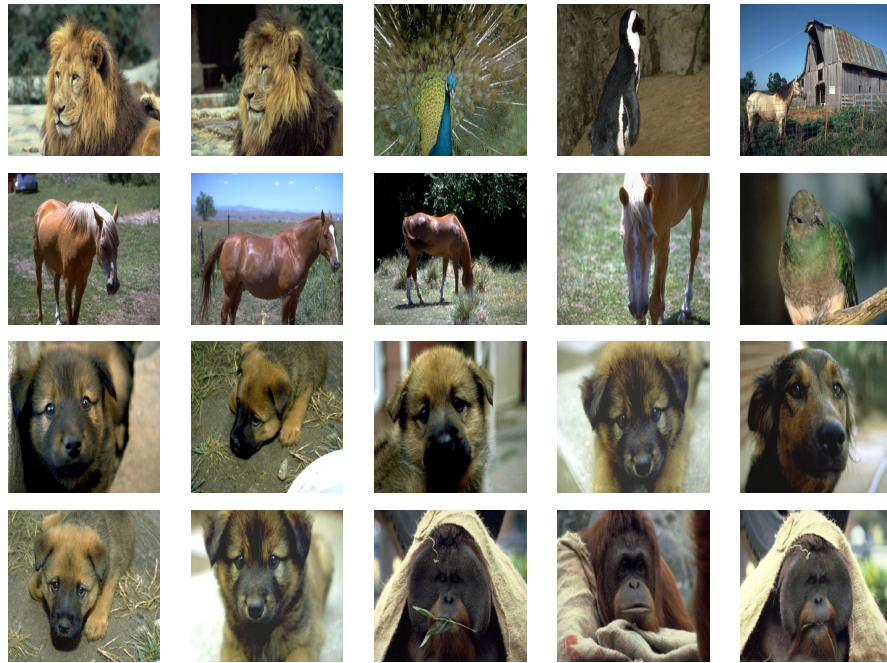


Table 5.4: First retrieval of the new query

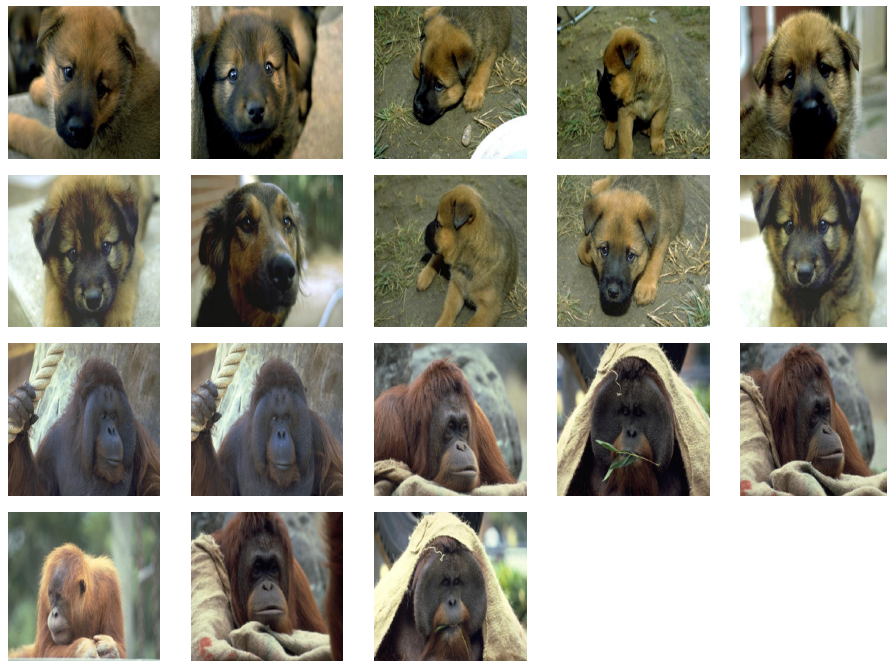


Table 5.5: Long term learning output of new query



Table 5.6: Modified Long term learning output of new query

Chapter 6

Conclusions and Future Work

6.1 Conclusions

A learning frame work was described which makes use of relevance feedback to enhance the performance of an image retrieval system from both short term learning and long term learning perspectives. The proposed long term learning scheme infers a semantic space from user's interactions. A method of updating the semantic space and guidelines for choosing the optimal dimensionality were also discussed. The learned semantic space supplements the low-level features in making image search result more satisfactory to the user. The semantic space is constructed by doing svd on H which only captures semantics arrived from relevance of an image. Hidden semantics of low-level features are ignored. To extract meaningful semantics method is modified by constructing the semantic space which include both low-level and high level features. It is seen that it will have more meaningful semantics since the facts of the images and the expert perceptions are coupled.

6.2 Future Work

Currently, each image belongs to one and only one semantic category, further research could analyze how results change when images belong to multiple semantic categories. One can also use gradations in constructing the semantic space instead of ones and zeroes and see the performance.

Bibliography

- [1] XiaofeiHe, OliverKing, Wei-Ying Ma, MingjingLi, Hong-Jiang Zhang, Learning a Semantic Space From Users Relevance Feedback for Image Retrieval. *IEEE Transactions CSVT, Vol.13, No.1, pp.39-48, jan2003*
- [2] M-Shan, J.Xiaojun Qi, Short term and Long-term learning approach for Content based image retrieval. *Proc IEEE Conference, volume: 2, pp: II-II, 2006*
- [3] Praks P., Dvorsky J., Snasel V, Latent Semantic Indexing for Image Retrieval System, *Proc. of the SIAM Conference on Applied Linear Algebra, Williamsburg, July 15-19, 2003.*
- [4] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, 41(6):391-407,1990*
- [5] Ritendra Datta and Dhiraj Joshi and Jia Li and James and Z. Wan, Image retrieval: Ideas, influences, and trends of the new age *ACM Computing Surveys, volume 39, pp .2007, 2006.*
- [6] T.Yoshizawa and H.Schweitzer. Longterm learning of semantic grouping from relevance feedback. *proc of Intl workshop on MIR pages,pages 165-172,NY,USA,2004.ACM*
- [7] N.Littlestone Learning quickly when irrelevant attributes abound : A new linear-threshold algorithm, *Mach.Learn.,vol.2,no.4,pp.285-318-1988*
- [8] I.Dagan,Y.Karov,and D.Roth,, “Mistake Driven learning in text categorization”, *in Proc.2ndConf.Empirical Methods in Natural Language Processing,1997,pp.55-63.*
- [9] W.Y.Ma and B.S.Manjunath, “Netra : A toolbox for navigating large image databases”, *ACM Multimedia Syst.,vol.7,pp.184-198,1999.*
- [10] YingLiu,DengshengZhang,GuojunLu,Wei-YngMa, “A survey of content based image retrieval with high-level semantics”, *Pattern Recognition,v.40 n.1,p.262-282,2007.*
- [11] Jiang, Wenchao and Wan, Baoping and Zhang, Qijun and Zhou, Yanhong, “Image Search by Latent Semantic Indexing Based on Multiple Feature Fusion” *CISP*

'08: *Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 2, 2008*.pp:515-519

- [12] Peter Praus,Pavel Praks,, “Automatic Retrieval with in water related pictures database using latent semantic indexing method” *ISSN Vol 2 pp:19-28-2009*.
- [13] Chun Che Fung and Kien-Ping Chung, “Establishing semantic Relationship in Inter-query Learning for Content Based Image Retrieval Systems” *PAKDD ,pp:498-506,2007*
- [14] BING WANG ,MEI-WU HE,SHUO WANG,MIAO WANG, “Image Retrieval using long term learning relevance feedback” *Proccedings of ML and Cybernetics,Hongkong,pp:19-22,2007*.
- [15] Donn Morission,Stephane Marchand-Maillet , Eric Bruno “Automatic Image Annotation with relevance feedback and Latent Semantic Analysis.” *Springer Berlin,vol:4918,pp:71-84,2008*.