

# Baseline of Content-Based Image Retrieval System

Shubhangi Durgi<sup>\*</sup>, Prof. Pravinkumar Badadapure<sup>\*\*</sup>

<sup>\*</sup>Electronics and telecommunication Department, Imperial college of Engineering, Pune, India

<sup>\*\*</sup>Electronics and telecommunication Department, Imperial college of Engineering, Pune, India

\*Shubhad10@gmail.com, \*\*badadapurep@gmail.com

**ABSTRACT**-Content-based image retrieval (CBIR) is used to identify relevant images based on their similarities to query images. This paper presents a novel framework for accurate image retrieval using three image features i.e. color, texture and shape. We use Color Histogram, Color Moment & Color Coherence methods for Color feature extraction. Texture is extracted using Tamura Feature. Corner detection is used for shape feature extraction. In this paper, we formulate two algorithms as Euclidean algorithm for linear structure & Image manifold algorithm for non-linear structure. We consider the problem of learning a mapping function from low-level feature space to high-level semantic space. Under the assumption that the data lie on a submanifold embedded in a high dimensional Euclidean space, we propose a relevance feedback scheme which is conducted only on the image manifold in question rather than the total ambient space. We then develop an algorithmic framework to approximate the optimal mapping function by a Radial Basis Function (RBF) neural network. We name the new algorithm as Image Manifold Learning (IML). The semantics of a new image can be inferred by the RBF neural network. Experimental results show that our approach is effective in improving the performance of content-based image retrieval systems.

**Index Item-** Color histogram, Color Moment, Color Coherence, Corner detection, Submanifold, Radial Basic Function Neural Network, Manifold learning.

## 1. INTRODUCTION

CBIR has been very challenging topic, because CBIR [1],[4],[11] is based on high level feature & low level feature. Low level features visualize color, texture, shape & so on. High level feature express emotions meaning association of feature expression with combination of perceptual feature. Thus, It is difficult to extract high level features like emotions, or what are the activities present in that image. But they give relatively more important meanings of objects and scenes in the images that are perceived by human beings. So generally low level features like color, texture, shape & edge are used for retrieval of the image.

Fig. 1 shows the architecture of a typical CBIR system. Each image in the image database is in standard form. For all images in database, first, features are extracted and the obtained feature space is stored in the feature database. When a query image is selected, its feature space will be compared with those in the feature database one by one and the similar images with the smallest feature distance will be retrieved.

CBIR can be divided into mainly two stages:

- **Preprocessing:** First step is to extract a feature, which describes its contents. In this processing, we perform feature filtration, normalization, segmentation (i.e. divide the image content - visual feature, emotion, color shape etc) and object identification. The output of this stage is a set of significant regions and objects.

- **Feature extraction:** Low level features are used to describe the content of the image. Image Features can be classified into primitives.

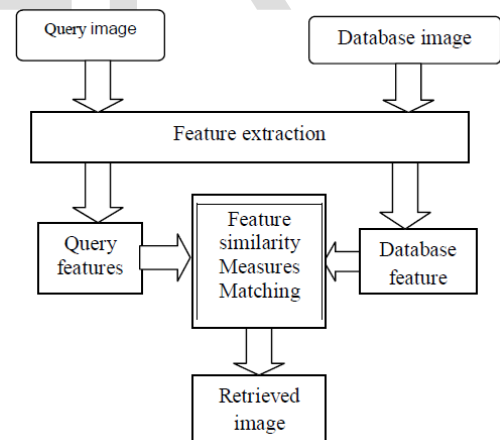


Fig. 1: Image Retrieval processing

In Euclidean distance, image retrieval techniques build on the assumption that the image space is Euclidean. However, in many cases, the image space might be a non-linear sub-manifold which is embedded in the ambient space. Intrinsically, there are two fundamental problems in image retrieval: 1) how do we represent an image? 2) How do we judge similarity?

One possible solution to these two problems is to learn a mapping Function from the low-level feature space to the

high-level semantic space [2]. The former is not always consistent with human perception while the latter is what image retrieval system desires to have. Specifically, if two images are semantically similar, then they are close to each other in semantic space. In this paper, our approach is to recover semantic structures hidden in the image feature space such as color, texture, shape etc

As we point out, the choice of the similarity measure is a deep question that lies at the core of image retrieval. In recent years, manifold learning [3],[6],[10],[11] has received lots of attention and been applied to face recognition[8], graphics[9], document representation [6], etc. These research efforts show that manifold Structure is more powerful than Euclidean structure for data representation, even though there is no convincing evidence that such manifold structure is accurately present. Based on the assumption that the images reside on a low-dimensional sub manifold, a geometrically motivated relevance feedback scheme is proposed for image ranking, which is naturally conducted only on the image manifold in question rather than the total ambient space

It is worthwhile to highlight several aspects of the framework of analysis presented here:

(1) Throughout this paper, we denote by image space the set of all the images. Different from most of previous geometry-based Works which assume that the image space is a Euclidean space [13], [11] in this paper, we make a much weaker assumption that the image space is a Riemannian manifold embedded in the feature space. Particularly, we call it image manifold. Generally, the image manifold has a lower dimensionality than the feature space. The metric structure of the image manifold is induced but different from the metric structure of the feature space. Thus, a new algorithm for image retrieval which takes into account the intrinsic metric structure of the image manifold is needed.

(2) Given enough images, it is possible to recover the image Manifold. However, if the number of images is too small, then any algorithm can hardly discover the intrinsic metric structure of the image manifold. Fortunately, in image retrieval, we can make use of user provided information to learn a semantic space that is locally isometric to the image manifold. This semantic space is Euclidean and hence the geodesic distances on the image manifold can be approximated by the Euclidean distances in this semantic space. This intuition will be strengthened in our experiments.

(3) There are two key algorithms in this framework. One is the retrieval algorithm on image manifold, and the other is an algorithm for learning a mapping function from feature space (color, texture, etc.) to high-level semantic space. The learning algorithm will gradually "flat" the image manifold, and make it better consistent with human perception. That is, if two images are close (in the sense of Euclidean metric) to each other, they are semantically similar to each other.

## 2. FEATURE EXTRACTION

Feature vector includes three color features and three textural features & shape feature, seven features in total. Color features are color histogram, color moments and color coherence. Textural features include Tamura contrast, Tamura directionality and Tamura coarseness & shape feature include corner detection.

### 2.1. Color Feature Extraction

#### 2.1.1 Color Histogram

A histogram is a way to approximate the distribution of a random variable. It is also a simple approach to give a description of an estimated density. To create a color histogram the color space has to be divided into regions. For example, the widely used 24 bit RGB color space contains 224 regions. A histogram containing as many histogram bins would be too large to be dealt with efficiently. To reduce the amount of memory needed the feature space is quantized. Here, it is required to find a good tradeoff between loss of precision and memory requirement. For gray images, the situation is somewhat better because gray images usually contain 256 different gray levels only. 256 bins are still a manageable amount of data. After partitioning the feature space, for each region the number of pixels from this region is counted to calculate the empirical probabilities. In the implementation of the color histogram calculation we used  $4^3$  bins:

**2.1.2 Color moments** - The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments (e.g. Normal distributions are differentiated by their mean and variance). It therefore follows that if the color in an image follows a certain probability distribution, the moments of that distribution can then be used as features to identify that image based on color. Typically, three central moments of a image's color distribution are used. They are Mean, Standard deviation and Skewness. A color can be defined by 3 or more values. (In this study we will restrict ourselves to the RGB scheme). Moments are calculated for each of these channels in an image. An image therefore is characterized by 9 moments - 3 moments for each 3 color channels. We will define the  $k$ -th color channel at the  $(i,j)$ -th image pixel as,  $I_{ij}(K)$ ,  $i=1...m$ ,  $j=1...n$ . The three color moments can then be defined

$$\text{Mean: } E_k = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n I_{ij}(k) \quad (1)$$

Mean can be understood as the average color value in the image.

$$\text{Standard deviation: } \sigma_k = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I_{ij}(k) - E_k)^2} \quad (2)$$

The standard deviation is the square root of the variance of the distribution.

$$\text{Skewness: } s_k = \sqrt[3]{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I_{ij}(k) - E_k)^3} \quad (3)$$

Skewness can be understood as a measure of the degree of asymmetry in the distribution.

### 2.1.3. Color Coherence Vector (CCV)

CCV is a more complex method than Color Histogram. It classifies each pixel as either coherent or incoherent. Coherent pixel means that it's part of a big of connected component (CC) while incoherent pixel is part of a small connected component. Of course first we define the criteria which we use to measure whether a connected component is big or not. To calculate CCV:

- Blur the image (by replacing each pixel's value with the average value of the 8 adjacent pixels surrounding that pixel).
- Discretize the color-space (images' colors) into n distinct color.
- Classify each pixel either as coherent or incoherent. This is computed by
  - Find connected components for each discretized color.
  - Determine tau's value (Tau is a user-specified value (Normally it's about 1% of image's size)).

Any Connected Component has number of pixels more than or equal to tau then its pixels are considered coherent and the others are incoherent.

- For each color compute two values (C and N).
  - C is the number of coherent pixels.
  - N is the number of incoherent pixels.

It's clear that the summation of all color's C and N = number of pixels.

### 2.2. Tamura textural features.

Tamura feature [14] is designed in accordance with psychological studies on the human perception of texture: coarseness, contrast, directionality, line-likeness, regularity, and roughness. They make experiments to test the significance of the features. They found the first three features to be very important. That is, these correlate strongly with the human perception. These three features, coarseness, contrast, and directionality, are defined as follows:

**2.2.1 Coarseness** - gives information about the size of the texture elements. The higher the coarseness value is, the rougher is the texture. If there are two different textures, one

macro texture of high coarseness and one micro texture of low coarseness, the macro texture is considered. The essence of calculating the coarseness value is to use operators of various sizes. A large operator is chosen when a coarse texture is present even if there is a micro-texture and a small operator is chosen when micro texture is present only.

**2.2.2 Contrast**- In the narrow sense, contrast stands for picture quality. More detailed, contrast can be considered to be influenced by the following four factors:

- Dynamic range of gray-levels
- Polarization of the distribution of black and white on the gray-level histogram
- Sharpness of edges
- Period of repeating patterns.

**2.2.3 Directionality**-Not the orientation itself but presence of orientation in the texture is relevant here. That is, two textures differing only in the orientation are considered to have the same directionality.

### 2.3 Shape feature

#### 2.3.1 Corner detection

The Harris Detector [16] is a commonly used method for extracting corner point locations.

$$E_{u,v}(x,y) = [u,v] \sum_{xy} W(x,y) H(x,y) \begin{bmatrix} u \\ v \end{bmatrix} = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (4)$$

The Harris method uses a weighting function  $w(x; y)$  for the window. Common functions for  $w(x; y)$  are step functions and Gaussians. Gaussian functions are popular because they treat the image data more symmetrically, rather than preferring directions parallel to either of the axes.

Since the goal of the corner detector is to identify when  $E_{u,v}$  varies in all directions, the next step is to associate a value to this amount. The eigenvalues of the matrix  $M$  are a good indication of this. When both eigenvalues are small there is little change for any  $(u; v)$ . When one is large and the other is small it indicates that there is an edge, since one direction has high change, while the orthogonal direction has small change. When both eigenvalues are large it indicates a corner.

The Harris Detector avoids computing the eigenvalues directly, and opts for a method of computation which does not require the square root operator:

$$C(M) = \det(M) + K * \text{tr}^2(M) \quad (5)$$

Due to the squared terms this function is large when both singular values are large.

### 3. EUCLIDEAN DISTANCES

Experimental database consist of 90 images of different colored flowers. All images are not in same size, so they need to be resized. First step is to select query image and then

extract the color feature of query image & database image. Find the distance between the feature of query & database image. To measure the distances we use Euclidean distances algorithm which is represented as

$$D = \sqrt{\sum_{i=1}^N [F_Q - F_{DB}(i)]^2} \quad (6)$$

$F_Q$  is the feature vector of query image;  $F_{DB}$  is feature vector of database images.

## 4. IMAGE MANIFOLD LEARNING ALGORITHM

### 4.1 Inferring a Distances Matrix in semantic Space from user Interaction

In this section, we describe how to infer a distance matrix in semantic space from user interactions. Some previous work could be found in [7]. Here, we present a simple method to update the distance matrix gradually.

Let  $B$  denote the distance matrix,  $B_{ij} = ||x_i - x_j||$ . Intuitively, the images marked by the user as positive examples in a query session share some common semantics. Therefore, we can shorten the distances between them. Let  $S$  denote the set of positive examples,  $S = \{s_1, s_2, \dots, s_k\}$ . We can adjust the distance matrix as follows

$$B_{S_{ij}} \rightarrow B_{S_{ij}} / \alpha \quad (7)$$

Where  $\alpha$  is a suitable constant greater than 1.

As the user interacts with the retrieval system, the distance matrix will gradually reflect the distances between the images in semantic space which is consistent with human perception.

### 4.2 Using Manifold Structure for Image Representation

In the above subsection, we have obtained a distance matrix in Semantic space. In this subsection, we discuss how to find the Semantic representation for each image in database, while the Distances are preserved. Recently, there has been some renewed interest [3][12][10] in the problem of developing low dimensional representations when data arises from sampling a probability distribution on a manifold. To choose a proper mapping algorithm, the following two requirements should be satisfied:

- 1) Since the image distribution in feature space is highly irregular and inconsistent with human perception, the mapping algorithm must have the locality preserving property.
- 2) The mapping algorithm should explicitly take into account the manifold structure.

Based on these two considerations, we use Laplacian Eigenmaps [1] to find such a mapping. We first compute the similarity matrix as follows

$$W_{i,j} = \begin{cases} \exp(-B_{i,j} / t), & B_{i,j} < \varepsilon \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

We found that  $t = 1$  and  $\varepsilon = 0.1$  provide good results.  $B$  is the distance matrix obtained in the previous subsection. Note that, the weight matrix has locality preserving property, which is the key feature of Laplacian Eigenmaps.

The objective function with our choice of weights  $W_{ij}$  incurs a heavy penalty if neighboring point's  $x_i$  and  $x_j$  are mapped far apart. Therefore, minimizing it is an attempt to ensure that if  $X_i$  and  $X_j$  is "close" then  $Y_i$  and  $Y_j$  is close as well. To minimize this objective function, it is equivalent to solve the following eigenvector problem:

$$Lf = \lambda Df \quad (9)$$

Where  $D$  is a diagonal matrix, whose entry is column sum (also row sum, since  $W$  is symmetric) of matrix  $W$ ,  $D_{ii} = \sum_j W_{ij}$ .  $L$  is called Laplacian matrix,  $L = D - W$ . Let  $y^{(0)}, y^{(1)}, \dots, y^{(n)}$  be the solutions of the above eigenvector problem, ordered according to their eigenvalues, It is easy to show that and  $y^{(0)} = (1, \dots, 1)$ . We leave out  $y^{(0)}$  and use the next  $k$  eigenvectors for embedding in  $k$ -dimensional Euclidean space.

$$X_i \rightarrow Z_i = (y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(k)}) \quad (10)$$

$Z_i$  is a  $k$ -dimensional map of image  $X_i$  in LE semantic spaces.

In summary, our goal is to find a vector representation (map) in semantic space for each image in database. Dimensionality reduction itself is not our goal, though we can make the dimensionality of the LE semantic space much lower than the feature space.

### 4.3 Radial Basic Function Neural Network

In the above section, every image in database is mapped into the semantic space. Now, the problem is that, for a new image outside the image database, it is unclear how to evaluate its map in the LE semantic space, since we don't have a mapping function. Here we present an approach that applies neural network to approximate the optimal mapping function, which intrinsically distinguishes our framework from previous work [7]. In this work, we use radial basis function (RBF) networks [15], and the standard gradient descent is used as a search technique. The mapping function learned by RBF networks can be represented by

$$F(x) = \sum_{i=1}^h w_{ij} G_i(x) \quad (11)$$

Where  $h$  is the number of hidden layer neurons,  $w_{ij} \in R$  are the weights.  $G_i$  is the radial function defined as follows:

$$G_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{\sigma_i^2}\right) \quad (12)$$

Where  $c_i$  is the center for  $G_i$ , and  $\sigma_i$  is the basis function width.

In summary, the RBF neural network approximates the optimal mapping function from low-level feature space to semantic space. It is trained off-line with the training samples  $\{x_i, z_i\}$ . The computational complexity in retrieval process



will be reduced as the dimensionality of the semantic space is reduced. The image representation  $f(\mathbf{x}_i)$  in RBFNN semantic space is an approximation of image representation  $\mathbf{Z}_i$  in LE semantic space, i.e.,  $f(\mathbf{x}_i) \cong \mathbf{Z}_i$  for a new image previously unseen, it can be simply mapped into the RBFNN semantic space by the mapping function  $f$ .

## 5. EXPERIMENTAL RESULTS

In this paper, we focus on image retrieval based on user's relevance feedback. The user can submit a query image either inside or outside the database. The system first computes low-level features of the query image and then maps it into semantic space using the learned mapping function. The system retrieves and ranks the images in the database. Then, the user provides his judgment of the relevance of retrieval. The image database we use consists of 90 images of 8 categories from the dataset. A retrieved image is considered correct if it belongs to the same category of the query image. Three types of color features (color histogram, color moment, color coherence), three types of texture features (tamura coarseness histogram, tamura directional), & corner detection for shape feature are used in our system.

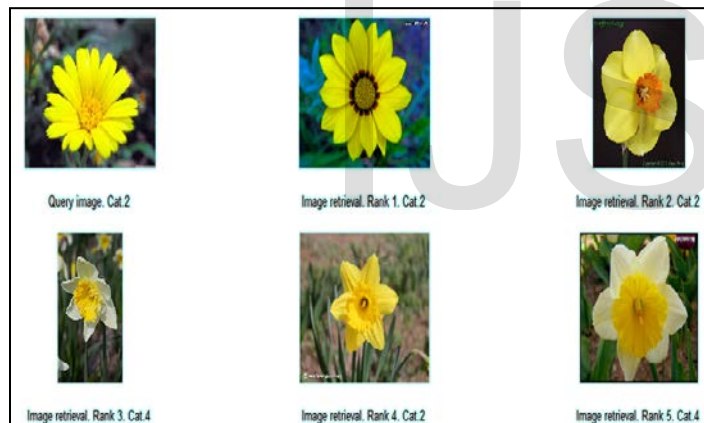


Fig 1. Retrieval of image- select query image inside the database by ED algorithm

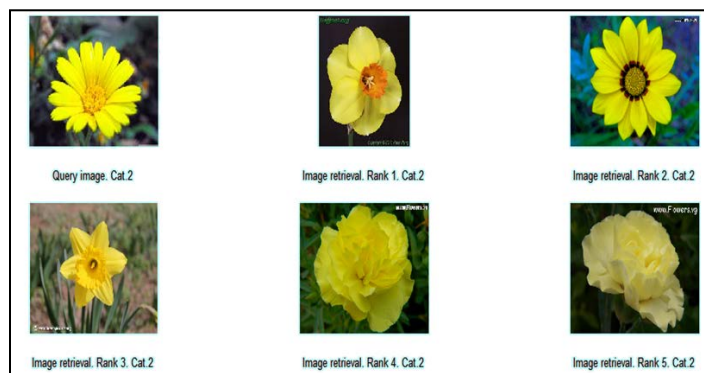


Fig 2. Retrieval of image- select query image inside the database by IML algorithm

Fig.1 & Fig. 2. Shows the retrieval results using ED & IML algorithm respectively when query image inside the database. Here we rank 5 images based on distance to the query image. As we see from result, retrieval using the IML algorithm provides better result compared to that of ED algorithm.

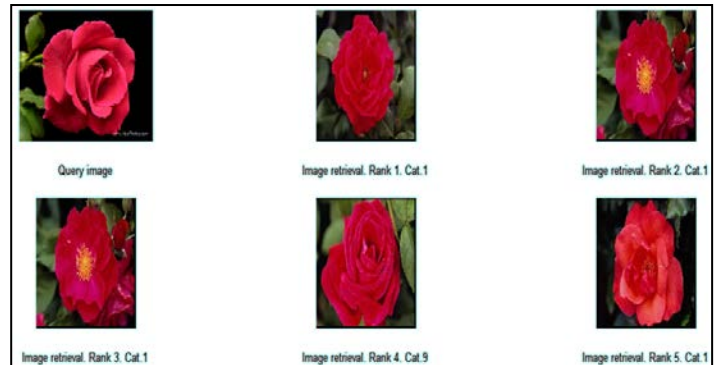


Fig 3. Retrieval of image- select query image outside the database by ED algorithm

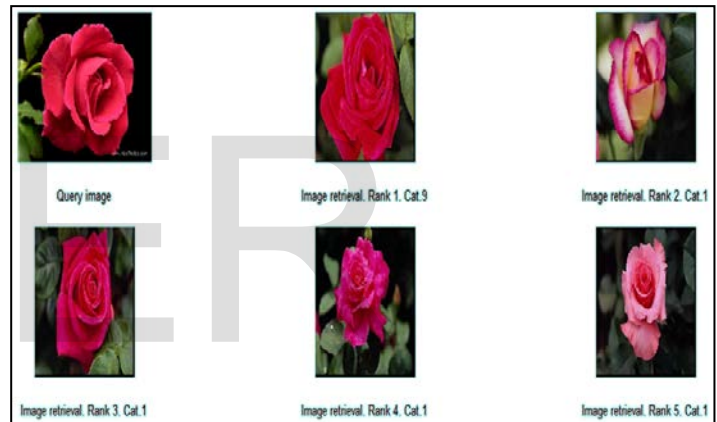


Fig 4. Retrieval of image- select query image outside the database by IML algorithm

Fig.3 & Fig. 4. Show the retrieval result using ED & IML algorithm respectively when query image outside the database. Here we rank 5 images based on distance to the query image.

Again, as we see from result, retrieval using the IML algorithm provides better result compared to that of ED algorithm. Thus IML algorithm outperforms ED algorithm in terms of the retrieval accuracy.

Fig 5 compares the retrieval accuracy of IML and ED algorithm for different dimensionalities. As seen from the plot above, accuracy of IML algorithm decreases as the dimensionality increases. Thus in our implementation we have reduced the dimensionality of feature vector for the better results. For ED algorithm there is no change in the accuracy as we consider full feature vector with all the dimensions. The retrieval accuracy is defined as follows:

$$Accuracy = \frac{\text{relevant images retrieval in top } N \text{ returns}}{N}$$

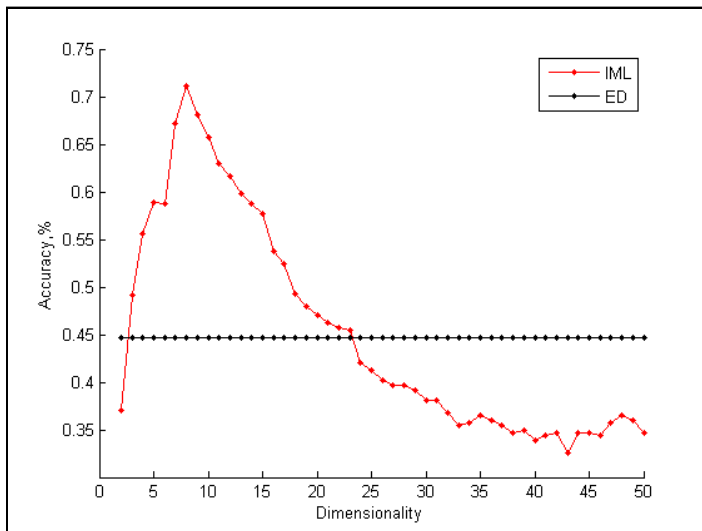


Fig 5. Retrieval accuracy comparison between IML & ED for different dimensionality.

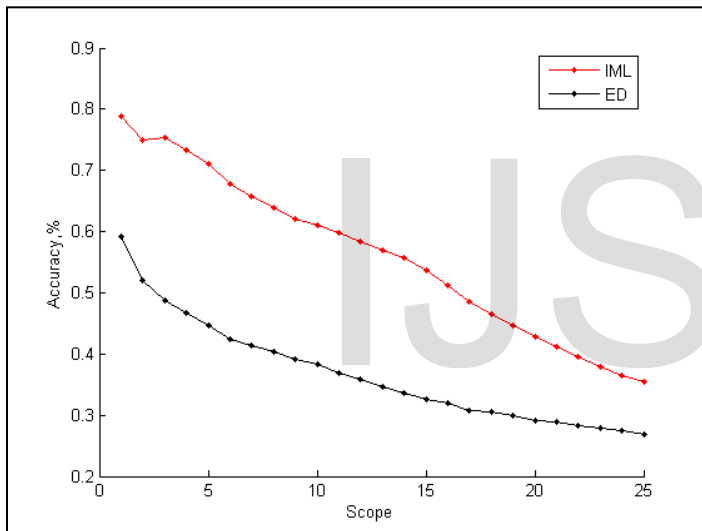


Fig 6. The retrieval accuracy comparison between IML & ED for different Scope

Fig. 6 shows the graph of retrieval accuracy vs. retrieval scope for both IML and ED algorithm. From the graph we can conclude that accuracy decreases as the retrieval scope increases. Again, IML algorithm outperforms the ED algorithm for retrieval accuracy. In our experiment we have kept the scope as 5 for better result

## 6. CONCLUSION

The proposed system is highly experimental and makes use of Euclidean Algorithm and color, shape & texture feature extraction to provide the solution for accurate image retrieval problem. Euclidean algorithm is used for distances metric learning; by this, we obtained retrieval accuracy in CBIR system.

In this paper, under the assumption that the data lie on a submanifold hidden in a high dimensional feature space, we developed an algorithmic framework to learn the mapping

between low-level image features and high-level semantics. It utilizes relevance feedback to enhance the performance of image retrieval system. This framework provides a solution to the two fundamental problems in image retrieval: how to judge similarity & how to represent an image.

To solve this problem, two semantic spaces, LE semantic space and RBFNN semantic space, are learned from user's relevance feedback. A mapping function is approximated by a RBF neural network. The semantic space gives a Euclidean representation of the image manifold.

In experimental result, image retrieval shows that IML algorithm gives better result than ED algorithm for both, query image which is inside & outside database image.

## REFERENCES

- [1] Luo Si, Rong Jin, Steven C. H. Hoi, and Michael R. Lyu, "Collaborative image retrieval via regularized metric learning" *Multimedia Syst.* 12(1): 34-44 (2006).
- [2] He, X., Ma, W.-Y., Zhang, H.-J., "Learning an image manifold for Retrieval" *In Proceedings of ACM MM 2004* (2004)
- [3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", *Advances in Neural Information Processing Systems*, 2001.
- [4] E. Chang, K. Goh, G. Sychay, and G. Wu, "CBSA: Content- Based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machine". *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, No. 1, Jan. 2003.
- [5] X. He and P. Niyogi, "Locality Preserving Projections", in *Advances in Neural Information Processing Systems 16, Vancouver, Canada*, 2003.
- [6] X. He, D. Cai, H. Liu and W.-Y. Ma, "Locality Preserving Indexing for Document Representation", in *ACM SIGIR conference on Information Retrieval, Sheffield*, 2004.
- [7] X. He, O. King, W.-Y. Ma, M.-J. Li, and H.-J. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval", *IEEE Trans. on Circuit and System for Video Technology*, Jan, 2003.
- [8] X. He, S. Yan, Y. Hu and H.-J. Zhang, "Learning a locality preserving subspace for visual recognition", in *Proc. IEEE Conf. on Computer Vision, Nice, France*, 2003.
- [9] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model", in *Proc. of SIGGRAPH*, 2003.
- [10] S.T. Roweis, and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", *Science*, vol 290, 22 December 2000.
- [11] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval", in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, June* 2000.
- [12] J.B. Tenenbaum, V.D. Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, Vol 290, 22 December 2000.
- [13] Y. Ishikawa, R. Subramanya and C. Faloutsos, "MindReader: query databases through multiple examples", *24<sup>th</sup> Conf. on Very Large Databases, New York*, 1998.
- [14] H. Tamura, S. Mori, T. Yamawaki, "Textural Features Corresponding to Visual Perception", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 6, pp. 460-472, June 1978.
- [15] Kui Wu & Kim-Hui Yip, "An Efficient Radial Basic Function Network Approach For Content-Based Image Retrieval" *IEEE* 2004.
- [16] C. Harris and M.J. Stephens, "A combined corner and edge detector", *Alvey Vision Conference*, (1988), 147152.

IJSER