

# Color & Texture Based Image Retrieval using Fusion of Modified Block Truncation Coding (MBTC) and Kekre Transform Patterns

A R Sawant  
PG Student, M. E. EXTC  
TCET, Kandivali  
Mumbai-400 101

Dr. V A Bharadi,  
Dr. H B Kekre  
Computer Department  
MPSTME, NMIMS University  
Mumbai-28

Bijith Markarkandy  
Deputy HOD, IT Department,  
TCET, Kandivali  
Mumbai-400 101

**Abstract** - Content Based Image Retrieval is an interesting topic of research. This paper is about image content-based image search. Specifically, it is on developing technologies for bridging the semantic gap that currently prevents wide-deployment of image content-based search engines. Mostly, content-based methods are based on low-level descriptions, while high-level or semantic descriptions are beyond current capabilities. In this paper a CBIR is proposed based on color & texture based search. Modified Block Truncation Coding (MBTC) is used for color information retrieval. To extract texture information we are using pattern generated by transforms, currently we are considering Kekre Transform. The feature vector is generated by fusion of above mentioned techniques.

**Keywords**- Image retrieval, CBIR, MBTC, Kekre's pattern.

## I. INTRODUCTION

Retrieval is the wide topic of research from the decades. The concept behind is to get desired data from the database. It may image, text, audio or video as per requirement of user. The basic types of retrievals are mentioned below, from ages images have been the mode of communication for human being. Today we are able to generate, store, transmit and share enormous amount of data because of the exhaustive growth of Information and Communication Technology. Much of this information is multimedia in nature, which consists of digital images, video, audio, graphics, and text data [1], [2]. But all that information is only useful if one can access it efficiently. This does not only mean fast access from a storage management point of view but also means that one should be able to find the desired information without scanning all information manually. Previous method used for image retrieval is Text based image retrieval. The advantage of textual indexing of image is that it can provide user with key word searching, catalogue browsing and even with query

interface. But the major drawback of text based image retrieval are, annotation depends on the person who adds it , the user of a Text Based Image Retrieval must describe an image using nearly the same keywords that were used by the annotator in order to retrieve that image [4]. Due to all these drawbacks, Content Based Image Retrieval is introduced.

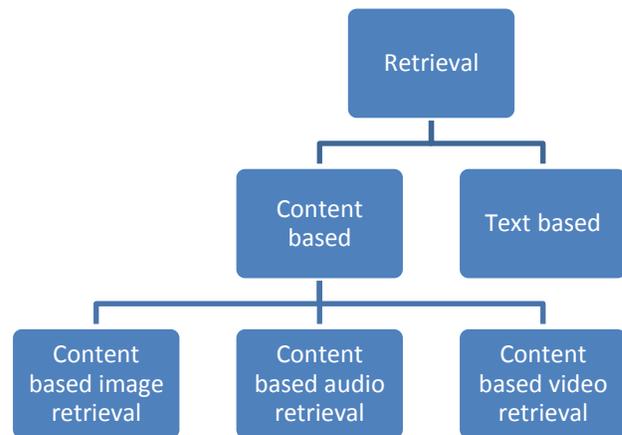


Figure 1. Types of retrieval

Here we will be dealing with features of images such as texture and color.

## II. CBIR EXISTING TECHNIQUES

H B Kekre, V A Bharadi et al. have introduced Content Based Image Retrieval using Fusion of Gabor Magnitude and Modified Block Truncation Coding [4]. Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction.

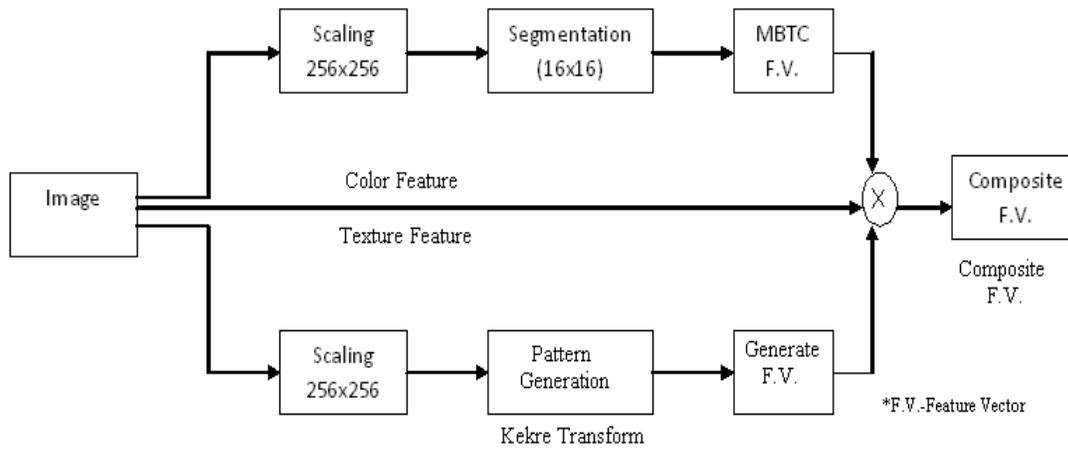


Figure 2. Basic Block Diagram for Planned Work

Expanding a signal using this basis provides a localized frequency description, therefore capturing local features/energy of the signal. Texture features can then be extracted from this group of energy distributions. And modified block truncation is used to retrieve color feature from image. They proved that the proposed system is giving higher Precision and Recall as compared to only Gabor and only MBTC based CBIR. Gabor feature gives good response to texture of the image and Modified BTC give good response to color content of image.

J. Zhang and W. Zou [5] have presented a novel technique that employs both the color and edge direction features for Content-Based Image Retrieval (CBIR). In this method, a given image is first divided into sub-block which has the same size and then the color and edge direction features of each sub block can be extracted. Next, it constructs a codebook of color feature using clustering algorithm and then each sub-block is mapped to the codebook. The color feature is used to retrieve images, and the edge direction feature is the weight of the similarity measure for the color feature.

Rose and Shah carried on a research project to improve the accuracy of CBIR Using Gradient Projections [6]; the image's structural properties were examined to distinguish one image from another. By examining the specific gray level of an image, a gradient can be computed at each pixel. Pixels with a magnitude larger than the thresholds are assigned a value of 1. These binary digits are added across the horizontal, vertical, and diagonal directions to compute three projections. These vectors are then compared with the vectors of the image to be matched using the Euclidean Distance Formula. These numbers are then stored in a bookmark so that the image needs only be examined once. A program has been developed for Matlab that performs this method of projecting gradients. Three databases were amassed for the testing of the proposed system's accuracy: 82 digital camera pictures, 1,000 photographic images, and a set of object orientated photos. The program was tested with 100% accuracy with all submitted images to the database, and was able to distinguish between pictures that fooled previous CBIR engines. The weakness of this project was its color-blindness.

A CBIR method based on color-spatial feature has been proposed by Lei, Fuzong & Zhang [7]. They proposed a fast algorithm which could include several spatial features of color in an image for retrieval because except for the color histogram information, the position information of each color plays an important role too. These features are area and position, which mean the zero-order and the first-order moments, respectively. By computing the moments of each color region the similarity of two images according to the weight of each factor can be computed. In fact, these features are a kind of representation for image in the scale of low resolution, and the sample image given by a user is usually a draft drawn by hand. Moreover, when a user judges the similarity between two pictures, he will firstly judge them in coarse scale. In this sense, this method is close to the vision model of our eyes. Because the features are simple and can be calculated in fast speed, better result can be made easily through training.

H B Kekre, S D Thepade et al. introduced image retrieval with Shape Features Extracted using Gradient Operators and Slope Magnitude Technique with Block Truncation Coding (BTC) [9] and tested on generic image database with 1000 images spread across 11 categories. The average precision and recall of all queries are computed and considered for performance analysis. Gradient operators used for shape extraction were Robert, Prewitt, Sobel and Canny which are known as 'Mask-Shape-BTC' CBIR techniques. The problem with these Mask-Shape-CBIR methods is the need of resizing the database images to match it with the size of query. This drawback is removed using proposed Mask-Shape-BTC-CBIR methods. In proposed image retrieval techniques the feature vectors are formed by applying the block truncation coding (BTC) on the shape image obtained using slope magnitude applied on gradient of the image in both horizontal and vertical direction.

H.B.Kekre, S. D. Thepade concentrated on more precise and faster retrieval techniques [10]. They had applied Walsh transform of different sizes on all images in database to extract the features and the feature vector database is generated. Then Walshlet feature vector of query image is compared with Walshlet feature vectors calculated for database images. The

proposed Gray- Walshlet and Color-Walshlet based CBIR techniques are tested using 55 queries fired on the image database with 1000 images spread over 11 categories. Among the different levels of Walshlet transform, Walshlet level-5 had given best results.

### III. PROPOSED TECHNIQUE

#### A. Modified Block Truncation Coding

Block truncation coding (BTC) is a relatively simple image coding technique developed in the early years of digital imaging. This method first divides the image into small non-overlapping image blocks. The small blocks are coded one at a time. For each block, the original pixels within the block are coded using a binary bitmap the same size as the original block and two mean pixel values [4].

The method first computes the mean pixel value of the whole block and then each pixel in that block is compared to the block mean. If a pixel is greater than or equal to the block mean, the corresponding pixel position of the bitmap will have a value of 1; otherwise it will have a value of 0. The simplest extension was to view a colour image as consisting of three independent grey scale images and apply BTC to each colour plane independently. Most colour images are recorded in RGB space, which is perhaps the most well-known colour space [11].

In modified BTC to create a binary bitmap in the RGB space, an inter-band average image (IBAI) is first created and a single scalar value is found as the threshold value.

First we will resize the image in 256\*256pixels. Now let  $X = \{r(i,j), g(i,j), b(i,j) \mid i = 1, 2, \dots, m. j = 1, 2, \dots, n\}$  be an  $m \times n$  color block in RGB space where  $m=n=256$ .

$$i_b = \frac{1}{3} r(i, j) + g(i, j) + b(i, j) \quad (1)$$

The threshold is computed as the mean of each colour,

$$T_1 = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m r(i, j) \quad (2)$$

$$T_2 = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m g(i, j) \quad (3)$$

$$T_3 = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m b(i, j) \quad (4)$$

The binary bitmap is computed as below,

$$bm_1(i, j) = \begin{cases} 1 & \text{if } r(i, j) \geq T_1 \quad \dots (5) \\ 0 & \text{if } r(i, j) < T_1 \end{cases}$$

$$bm_2(i, j) = \begin{cases} 1 & \text{if } g(i, j) \geq T_2 \quad \dots (6) \\ 0 & \text{if } g(i, j) < T_2 \end{cases}$$

$$bm_3(i, j) = \begin{cases} 1 & \text{if } b(i, j) \geq T_3 \quad \dots (7) \\ 0 & \text{if } b(i, j) < T_3 \end{cases}$$

After the creation of the bitmap, two representative (mean) colors are then computed. The two mean colors are  $MC1 = \{Cr1, Cg1, Cb1\}$  and  $MC2 = \{Cr2, Cg2, Cb2\}$ . Where upper-mean is calculated by,

$$C_{r_1} = \frac{1}{\sum_{i=1}^n \sum_{j=1}^m b_{m1}(i, j)} \sum_{i=1}^n \sum_{j=1}^m b_{m1}(i, j) r(i, j) \quad (8)$$

$$C_{g_1} = \frac{1}{\sum_{i=1}^n \sum_{j=1}^m b_{m2}(i, j)} \sum_{i=1}^n \sum_{j=1}^m b_{m2}(i, j) g(i, j) \quad (9)$$

$$C_{b_1} = \frac{1}{\sum_{i=1}^n \sum_{j=1}^m b_{m3}(i, j)} \sum_{i=1}^n \sum_{j=1}^m b_{m3}(i, j) b(i, j) \quad (10)$$

And lower-mean calculated as below,

$$C_{r_2} = \frac{1}{n \times m - \sum_{i=1}^n \sum_{j=1}^m b_{m1}(i, j)} \sum_{i=1}^n \sum_{j=1}^m (1 - b_{m1}(i, j)) r(i, j) \quad (11)$$

$$C_{g_2} = \frac{1}{n \times m - \sum_{i=1}^n \sum_{j=1}^m b_{m2}(i, j)} \sum_{i=1}^n \sum_{j=1}^m (1 - b_{m2}(i, j)) g(i, j) \quad (12)$$

$$C_{b_2} = \frac{1}{n \times m - \sum_{i=1}^n \sum_{j=1}^m b_{m3}(i, j)} \sum_{i=1}^n \sum_{j=1}^m (1 - b_{m3}(i, j)) b(i, j) \quad (13)$$

#### B. Pattern Generation for Texture

Patterns are the texture combinations generated from transform basis functions.

Number of patterns can be generated using transform matrices namely 4 pattern, 16 pattern, 64 pattern.  $N \times N$  matrix can be used to generate  $N^2$  patterns. For example, if we want to generate 16 pattern then  $4 \times 4$  matrix need to be used. Element wise multiplication of each row of the transform matrix is taken with all possible rows of the same matrix.

The 16 Kekre texture patterns [12] are generated using Kekre transform matrices of size  $4 \times 4$ . Fig. 3 gives  $2 \times 2$  kekre matrix and generation of four Kekre texture patterns.  $2 \times 2$  Kekre transform matrix is shown in figure, each row of this matrix is considered one at a time and is multiplied with all rows of the same matrix to generate Kekre texture patterns as shown in figure 4.

The  $4 \times 4$  Kekre transform matrix is given in figure 5 and visualization of 16 Kekre transform patterns generated using it is shown in figure 9, where black and grey colour scaled between 1 to 256 in the pattern and 0 is represented by white colour.

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Figure 3.  $2 \times 2$  Kekre Matrix

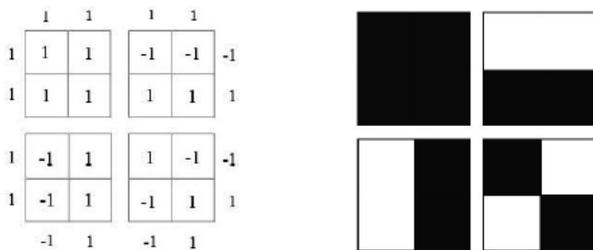


Figure 4. 4-Kekre Pattern Generation

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -3 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Figure 5.  $4 \times 4$  Kekre Matrix

### C. Fusion of MBTC & Patterns

1. We create a database containing images, the images are registered in the database. These images will be forwarded to Feature Vector Module i.e. fusion of MBTC and patterns generation. The resulted feature vector will be stored in feature vector database.
2. **Feature vector generation:**
  - (a) MBTC is applied on each image to calculate uppermean and lowermean.
  - (b) Uppermean and lowermean are 3-dimensional matrix i.e. it has three components red, green, blue. Three components are separated from uppermean and lowermean matrix.
  - (c) Each component is quantized to equivalent value of pattern matrix.
  - (d) Then occurrence of each pattern in six components i.e. uppermean(R,G,B) and lowermean (R,G,B) matrix is counted and arranged as  $(8 \times 12)$  matrix for each image in database. This matrix is Feature Vector.
3. Using above procedure feature vector is also calculated for query image.
4. Now feature vector of query image is compared with feature vector of images in database using Euclidian distance concept.
5. Then Euclidian distance is arranged in ascending order and according to Euclidian distances images are retrieved as result.
6. Images are then retrieved according to different thresholds (thresholds for Euclidian distances) and precision, recall calculated for each threshold value. Where,

$$\text{Precision} = \frac{\text{Total number of retrieve relevant images}}{\text{Total number of retrieve images}}$$

$$\text{Recall} = \frac{\text{Total number of retrieve relevant images}}{\text{Total number of relevant images in database}}$$

## IV. RESULT AND DISCUSSIONS

For each image red, green and blue components are separated from uppermean and lowermean and quantized in pattern values as shown in Fig. 6, Using  $4 \times 4$  Kekre matrix 16-pattern transform generated as shown in Fig 7.

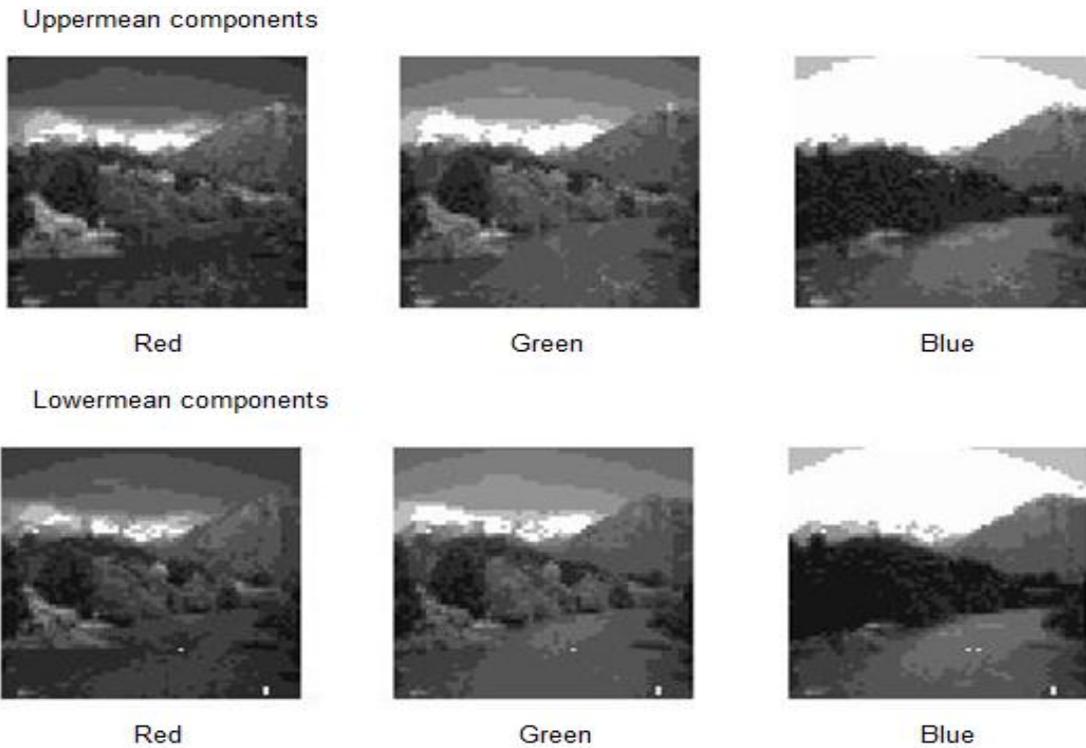


Figure 6. Quantized upperman and lowermean matrix

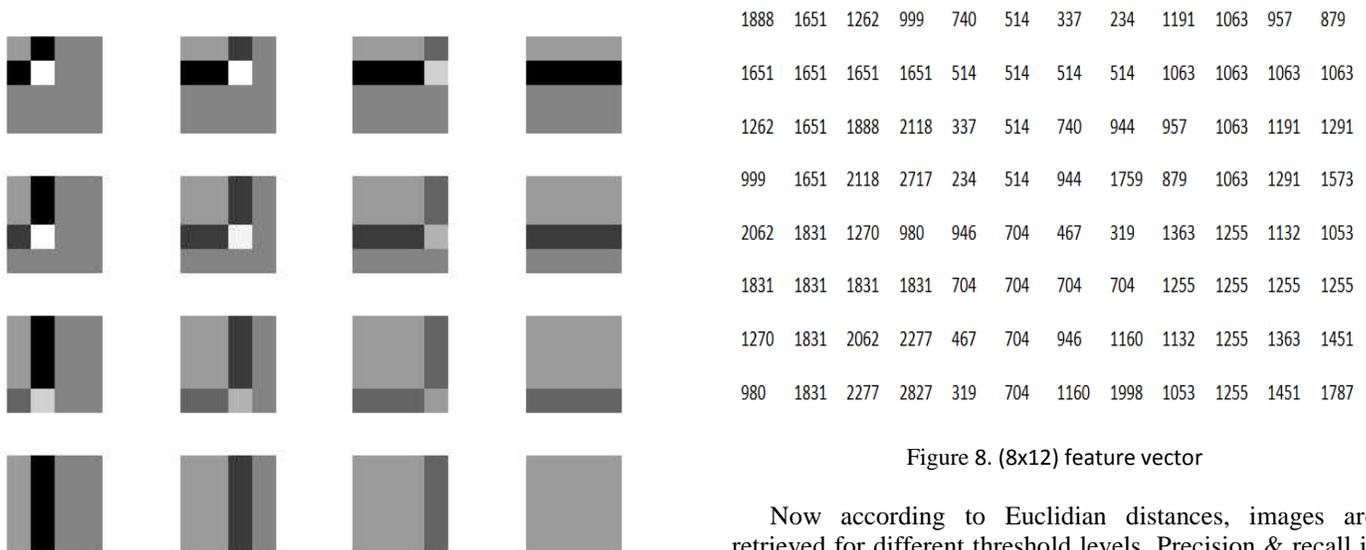


Figure 8. (8x12) feature vector

Figure 7. 16-Kekre Pattern Generation

Find co-occurrence of each pattern shown in figure is checked with components in figure. This co-occurrence matrix is nothing but feature vector as show in figure 8,

Now according to Euclidian distances, images are retrieved for different threshold levels. Precision & recall is calculated for three query images from different classes as shown below in Fig. 9 , 10 & 11. Fig. 12 shows a simple snapshot of the images retrieved for a given query image. The Recall and Precision curves are plotted, the crossover point is important as it gives the threshold at which precision & recall are optimal (Both are at maximum). The crossover range max 25% and min 20%. The results can be further improved by adding localized information to the feature vectors. We can segment the image and extract the feature vector for the segments.

## V. CONCLUSION

In this paper we have proposed a Content based Image Retrieval System based on Modified BTC & Kekre Transform pattern. MBTC extracts the color information and the Kekre Pattern is extracting the texture information. The feature vector is generated by a novel fusion mechanism.

The algorithm is tested on 1000 image database and we have achieved 10-25% Precision Recall crossover. The proposed technique is working satisfactory and can be further improved by adding localization to the feature vector.

## VI. REFERENCES

- [1] W.M. Smeulders, "Content-Based Image Retrieval at the End of the Early Years", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 12, DECEMBER 2000
- [2] G. Rafiee, S.S. Dlay, and W.L. Woo, "A Review of Content-Based Image Retrieval", IEEE, 2010. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.
- [3] Y N Mamatha and A.G Ananth and S O Neil, "Content Based Image Retrieval of Satellite Imageries Using Soft Query Based Color Composite Techniques", IEEE Trans on Acoustic speech signal processing, Vol 1, No.3, pp. 1278- 1288, 1986
- [4] Kekre H.B., Bharadi, V.A., Thepade S.D., Mishra B.K., Ghosalkar, S.E., Sawant S.M., "Content Based Image Retrieval Using Fusion of Gabor Magnitude and Modified Block Truncation Coding", IEEE computer society, 2010 IEEE
- [5] J. Zhang and W. Zou, "Content-Based Image Retrieval Using Color and Edge Direction Features", 2010 IEEE
- [6] Jed Rose and Mubarak Shah, "Content-Based Image Retrieval Using Gradient Projections", 1998 IEEE
- [7] Zhang Lei, Lin Fuzong, Zhang Bo, "A CBIR method based on color-spatial feature", 1999 IEEE
- [8] Zhao Hai-ying, Xu Zheng-guang, Penghong, "A Texture Feature Extraction Based On Two Fractal Dimensions for Content\_based Image Retrieval", 2008 IEEE
- [9] Dr.H.B.Kekre, S.D. Thepade et al., "Image Retrieval with Shape Features Extracted using Gradient Operators and Slope Magnitude Technique with BTC", International Journal of Computer Applications (0975 – 8887) Volume 6– No.8, September 2010
- [10] Dr.H.B.Kekre, S. D. Thepade, "Image Retrieval using Color-Texture Features Extracted from Walshlet Pyramid", ICGST - GVIP Journal, ISSN: 1687-398X, Volume 10, Issue 1, February 2010
- [11] H B Kekre and V A Bharadi, "Modified BTC & Walsh coefficients Based Features for Content Based Image Retrieval" NCICT, India.
- [12] Dr.H.B.Kekre, S.D. Thepade et al., "Performance Comparison of Gradient Mask Texture Based Image Retrieval Techniques using Walsh, Haar and Kekre Transforms with Image Maps", International Conference on Technology Systems and Management (ICTSM) 2011

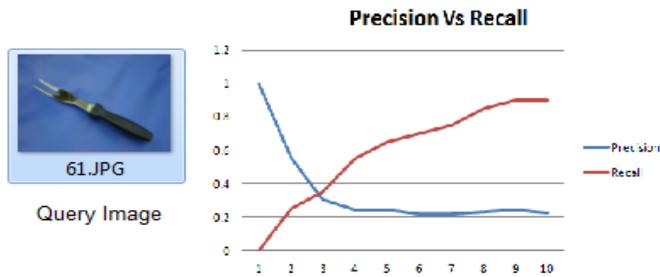


Figure 9. Precision-recall for class 1 (Crossover : 25%)

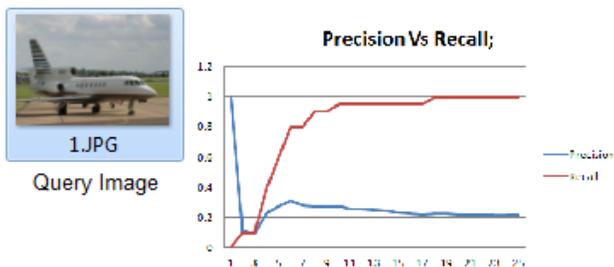


Figure 10. Precision-recall for class 2 Crossover : 10%

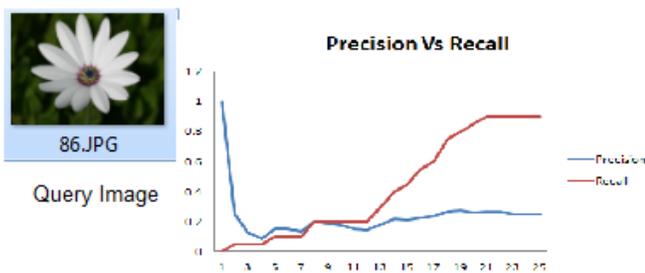


Figure 11. Precision-recall for class 3 Crossover : 20%

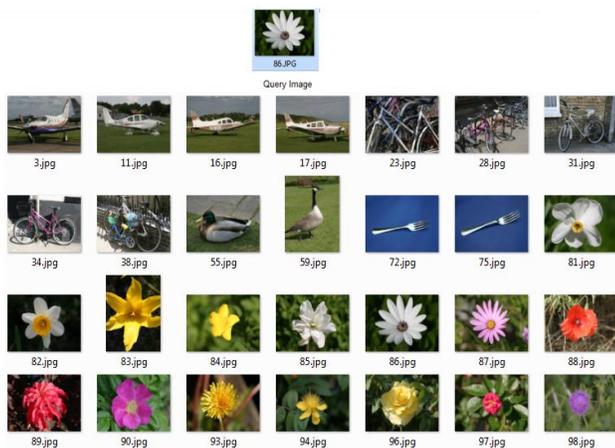


Figure 12. Sample Query Image & Retrieved Images