ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

CBIR USING COLOR, TEXTURE AND DISCRETE WAVELET TRANSFORM

[1] R.Venkat Ramana, [2] B.Prasuna

[1]Associate Professor, [2]Assistant Professor
Department of Electronics and Communication Engineering
Moghal College Of Engineering And Technology,Jntuh
Hyderabad,Telangana,India,
[1] Venkat_9827@yahoo.com

***Abstract*—The most content-based image retrievals (CBIR) systems use color for image retrieval. However, unsatisfactory results are often provided when image retrieval use only color features because in many cases, images with similar colors do not have similar content. As the solution of this problem is described as a novel algorithm for CBIR based on Color Edge Detection and Discrete Wavelet Transform (DWT). This method is different from the existing histogram based methods. This algorithm generates feature vectors that combine both color and edge features and also uses wavelet transform to reduce the size of the feature vector and simultaneously preserving the content details. The robustness of the system is also tested against query image alterations such as geometric deformations and noise addition etc. Further in this paper GLCM features are added as additional features to the existing wavelets and color edge featues which has enhanced the recognition accuracy. The performance of the system is evaluated and it is observed the precision has improved.**

***Keywords:-*—RGB, YCbCr, Color Edge Detection, Discrete wavelet transform, GLCM.**

1.INTRODUCTION

With the advancement in the internet and increase in the image records, retrieving the images from a large database became one of the most attractive research area. A lot of research has been done on Image Retrieval during last decade[1].To give text annotations manually to very large number of image records is become very tedious and impractical and has generated the need of an efficient image retrieval system. Content-based image retrieval measures the visual similarity between a query image and database images, Visual contents, commonly called as features are used by CBIR to search images from large scale image databases according to the requests of the user which is provided in the form of a query image[3]. The main advantage of CBIR is that it does not suffer from the subjectiveness of textual description. CBIR has diverse applications in internet, multimedia, medical image archives, crime prevention, entertainment, and digital libraries and it is an important field in image.

2.LITERATURE SURVEY

Color Feature

Several methods for retrieving images on the basis of color similarity have been described in the literature, but most are variations on the same basic idea. Each image added to the collection is analyzed to compute a *color histogram* which shows the proportion of pixels of each color within the image. The color histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each color (75% olive green and 25% red, for example), or submit an example image from which a color histogram is calculated. Either way, the matching process then retrieves those images whose color histograms match those of the query most closely. Variants of this technique are now used in a high proportion of current CBIR systems. Further cumulative color histograms are also proposed combining histogram intersection with some element of spatial matching and the use of region-based color querying. The results from some of these systems can look quite impressive but the process seems to be cumbersome.

Texture retrieval

The ability to retrieve images on the basis of texture similarity may not seem very useful. But the ability to match on texture similarity can often be useful in distinguishing between areas of images with similar color (such as sky and sea, or leaves and grass). A variety of techniques has been used for measuring texture similarity; the best-established rely on comparing values of what are known as *second-order statistics* calculated from query and stored images. Essentially, these calculate the relative brightness of selected *pairs* of pixels from each image. From these it is possible to calculate measures of image texture such as the degree of *contrast, coarseness*, *directionality* and *regularity*, or *periodicity, directionality* and *randomness*. Alternative methods of texture analysis for retrieval include the use of Gabor filters and fractals. Texture queries can be formulated in a similar manner to color queries, by selecting examples of desired textures from a palette, or by supplying an example query image. The system then retrieves images with texture measures most similar in value to the query. A recent extension of the technique is the texture thesaurus which retrieves textured regions in images on the basis of similarity to automatically-derived codeword's representing important classes of texture within the collection.

Shape retrieval

The ability to retrieve by shape is perhaps the most obvious requirement at the primitive level. Unlike texture, shape is a fairly well-defined concept – and there is considerable evidence that natural objects are primarily recognized by their shape. A number of features characteristic of object shape (but independent of size or orientation) are computed for every object identified within each stored image. Christo Ananth et al. [4] proposed a system which uses intermediate features of maximum overlap wavelet transform (IMOWT) as a pre-processing step. The coefficients derived from IMOWT are subjected to 2D histogram Grouping. This method is simple, fast and unsupervised. 2D histograms are used to obtain Grouping of color image. This Grouping output gives three segmentation maps which are fused together to get the final segmented output. This method produces good segmentation results when compared to the direct application of 2D Histogram Grouping. IMOWT is the efficient transform in which a set of wavelet features of the same size of various levels of resolutions and different local window sizes for different levels are used. IMOWT is efficient because of its time effectiveness, flexibility and translation invariance which are useful for good segmentation results.

Shape matching of three-dimensional objects is a more challenging task – particularly where only a single 2-D view of the object in question is available. While no general solution to this problem is possible, some useful inroads have been made into the problem of identifying at least some instances of a given object from different viewpoints. One approach has been to build up a set of plausible 3-D models from the available 2-D image, and match them with other models in the database. Another is to generate a series of alternative 2-D views of each database object, each of which is matched with the query image. Related research issues in this area include defining 3-D shape similarity measures, and providing a means for users to formulate 3-D shape queries.

3.PROPOSED SYSTEM

In this paper, a database of feature vectors has been prepared from images using Color Edge Detection and Wavelet techniques. For extraction of Color features, YCbCr and RGB color spaces are used. Bitmap images use the RGB planes directly to represent color images. But medical research proved that the human eyes have different sensitivity to color and brightness. The eye is more sensitive to changes in brightness than changes in color. Thus there came about the transformation of RGB to YCbCr [13]. The Y in YCbCr denotes the luminance component and Cb and Cr represent the chrominance components. The difference between YCbCr and RGB is that the YCbCr represents color as brightness and two color difference signals while the RGB represents color as red, green and blue. The YCbCr space was chosen because the luminance component Y is independent of the color so can be adopted to solve the color variation problem [14]. For extracting edges, Canny edge detection is used. Canny edge detector is an optimal detector. The detector ensures only one response to a single edge and it provides shape that is optimal at any scale. Canny edge detector is also able to cope up with noise in the image [15]. When the query image comes, a combined feature vector is computed for color and edge features. If the distance between feature vector of the query image and images in the database is small enough, the corresponding images in the database are to be selected as a match to the query image. The search is usually based on similarity rather than on exact match and the retrieval results are then ranked accordingly to a similarity index, for similarity measure Manhattan distance has been used.

.

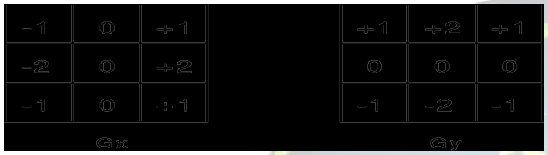
4.COLOR EDGE DETECTION

Contouring, segmentation and recognition of objects rely on a precise edge detection or edge enhancement. Edges evidence the structure and shape of objects as well as fine details of an image, i.e., they correspond to the high spatial frequencies of the Fourier spectrum of the image. Therefore, edge enhancement increases the discrimination capability of the segmentation and recognition systems. Edges in gray scale images are defined in an achromatic way as discontinuities in brightness function. Hence, in this kind of images, edge detection is accomplished by means of searching rapid changes in intensity values. Besides considering an extension of this procedure to color images, color edge detection also involves finding discontinuities along the adjacent regions of a color image in a certain 3D color space, e.g. RGB- or HSI-color space, that would capture different color characteristics [3,4]. Both changes in brightness and color between neighboring pixels should be exploited for more efficient color-edge extraction. It is found that 10% of the edges detected in color images fall into this non-intensity class.

Edge Detection Tecniques :

a)Sobel Operator

The operator consists of a pair of 3×3 convolution kernels as shown in Fig. 1. One kernel is simply the other rotated by 90°.



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

(i)Simple 3x3 matrix (ii) 90°rotated 3x3 matrix

Fig. 1: A pair of 3×3 convolution kernels

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|Gx| = \sqrt{(Gx^2 + Gy^2)}$$

Typically, an approximate magnitude is computed using:

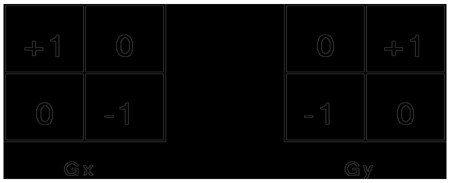
$$|G| = |Gx| + |Gy|$$

which is much faster to compute. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(Gy/Gx)$$

b)Robert's cross operator:

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The operator consists of a pair of 2×2 convolution kernels as shown in Figure. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.



+1	0
0	-1

Gx

0	+1
-1	0

Gy

(i)Simple 2x2 matrix (ii) 90°rotated2x2 matrix

Fig. 2: A pair of 2×2 convolution kernels

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|Gx| = \sqrt{(Gx^2 + Gy^2)}$$

although typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much faster to compute. The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan\left(\frac{Gy}{Gx}\right) - 3\pi / 4$$

c)Prewitt's operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

$$h1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

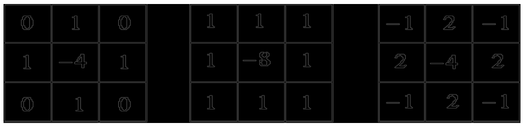
d)Laplacian of Gaussian:

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single graylevel image as input and produces another gray level image as output.

The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by:

$$\mathrm{L(x,y)} = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in Fig. 1.



0	1	0		1	1	1		−1	2	−1
1	−4	1		1	−8	1		2	−4	2
0	1	0		1	1	1		−1	2	−1

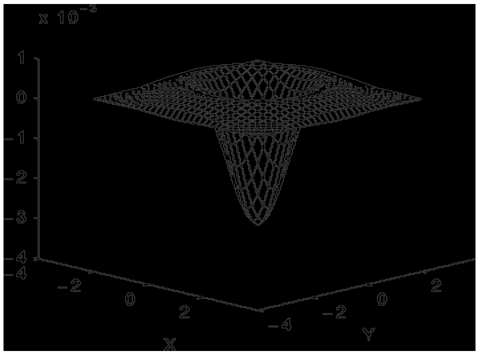
Fig..4 Commonly used discrete approx. to Laplacian filter

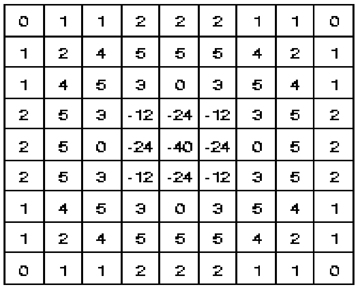
Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:

- Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.
- The LoG (`Laplacian of Gaussian') kernel can be recalculated in advance so only one convolution needs to be performed at run-time on the image.
- The 2-D LoG function centered on zero and with Gaussian standard deviation ■ has the form:

$$L_0 G(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$





0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Fig 5:Discrete approx. to LoG function with Gaussian ■= 1.4

Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels shown in Figure 1. This is because smoothing with a very narrow Gaussian (■ < 0.5 pixels) on a discrete grid has no effect. Hence on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussians.

5.CANNY'S EDGE DETECTION ALGORITHM

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a nonedge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

Step 1

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown below.

$\frac{1}{115}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Fig.6.Implementation using Gaussian mask

Step 2

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

-1	0	+1
-2	0	+2
-1	0	+1

Gx

(i)x-direction

+1	+2	+1
0	0	0
-1	-2	-1

Gy

(ii) y-direction

Fig. 7: A pair of 3×3 convolution mask

The magnitude, or edge strength, of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy|$$

Step 3

The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sumX is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

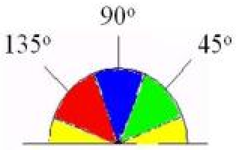
Theta = invtan (Gy / Gx)

Step 4

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x x x x x
x x x x x
x x a x x
x x x x x
x x x x x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.



Therefore, any edge direction falling within the **yellow range** (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the **green range** (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the **blue range** (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the **red range** (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5

After the edge directions are known, nonmaximum suppression now has to be applied. Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

Comparison of Edge detection Algorithm:

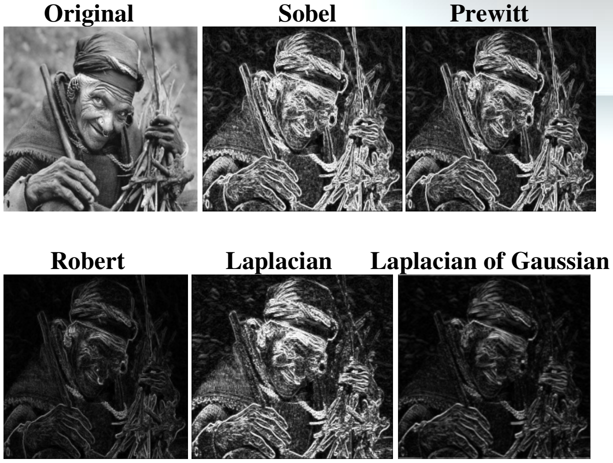


Fig.8. Comparison of Edge detection Algorithm

6.GREY LEVEL CO-OCCURRENCE MATRIX

The role of automatic visual inspection has grown up during the last decade in the same way as the computational power has increased and the prize of computation has decayed. Visual inspection, especially, in the sense of visual quality control has gained in popularity. Nowadays, several commercial applications are available e.g. in paper mills in saw mills, and in rock industry. Most of these applications are based on texture classification. Visual texture contains variations of intensities, which form certain repeated patterns. Those patterns can be caused by physical surface properties, such as roughness, or they could result from reflectance differences, such as the color on a surface. Differences observed by visual inspection are difficult to define in quantitative manner, which leads to demand of defining texture using some features.

Among the current approaches used in image processing to describe texture, the so called statistical approach is the widely used because it produces good results with low computational costs. This method considers the distribution of gray levels and their interrelationship. The pixel values are used to construct numerical structures which are associated to the texture pattern of the image. This pattern is based mainly on the inter-relationship between one pixel and its neighbors.

Grey Level Co-Occurrence Matrix (GLCM) is one of the best known texture Analysis methods. GLCM estimates image properties related to second order statistics. Each entry (i,j) in GLCM corresponds to the number of occurrences of the pair of grey levels I and j which are a distance d apart in the original image.

Grey Level Co-Occurrence Matrices are of two types:

First order GLCM

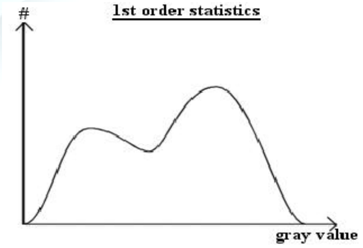


Fig.9: 1st order GLCM

Second order GLCM

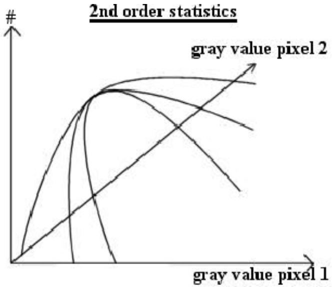


Fig10: 2nd Order GLCM

The pixel pairs (i,j) are defined by distance and angle:

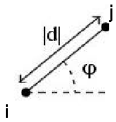


Fig11: Distance, Angle Relationship

Mathematical Framework of GLCM

A co-occurrence matrix is a square matrix with elements corresponding to the relative frequency of Occurrence of pairs of gray level of pixels separated by a certain distance in a given direction. Formally, the elements of a GxG gray level co-occurrence matrix Pd for a displacement vector d = (dx,dy) is defined as :

Pd (i,j) = |{((r,s) , (t,v)) :I (r, s)=i, I (t, v)= j}|

where I (×,×) denote an image of size NxN with G gray values, (r, s), (t, v)Î NxN, (t,v)=(r + dx, s + dy) and |. | is the cardinality of a set.

Haralick, Shanmugan and Dinstein proposed 14 measures of textural features which are derived from the co-occurrence matrices, and each represents certain image properties as coarseness, contrast, homogeneity and texture complexity.

Those that most commonly used are given below:

a)Entropy :

Entropy measures the disorder of an image and it achieves its largest value when all elements in P matrix are equal [3]. When the image is not texturally uniform many GLCM elements have very small values, which imply that entropy is very large. Therefore, entropy is inversely proportional to GLCM energy.

$$ENT = -\sum_i \sum_j P(i,j) log P(i,j)$$

b)Contrast:

Contrast feature is a measure of the image contrast or the amount of local variations present in an image.

$$CON = -\sum_i \sum_j (i-j)^2 P(i,j)$$

c)Angular Second Moment:

Energy, also called Angular Second Moment and Uniformity is a measure of textural Uniformity of an image. Energy reaches its highest value when grey level distribution has either a constant or a periodic form. A homogenous image contains very few dominant grey tone transitions and therefore the matrix for this image will have fewer entries of larger magnitude resulting in large value for energy feature. In contrast if the matrix contains a large number of small entries, the energy feature will have a smaller value.

d)Inverse Difference Moment:

Inverse difference moment measures image homogeneity. This parameter achieves its largest value when most of the occurrences in GLCM are concentrated near the main diagonal. IDM is inversely proportional to GLCM contrast.

$$IDM = \sum_i \sum_j \frac{1}{1+(i-j)^2} P(i,j)$$

In all the above equations p(i,j) refers to the normalized entry of the co-occurrence matrices. That is p(i,j) = Pd (i,j)/R where R is the total number of pixel pairs (i,j). For a displacement vector d = (dx,dy) and image of size NxM R is given by (N-dx)(M-dy).

7.Results

The benchmarks allow the comparison of image retrieval systems under different aspects: usability and user interfaces, combination with text retrieval, or overall performance of a system. The WANG database is a subset of 1,000 images of the Corel stock photo database which have been manually selected and which form 10 classes of 100 images each. The WANG database can be considered similar to common stock photo retrieval tasks with several images from each category and a potential user having an image from a particular category and looking for similar images which have e.g. cheaper royalties or which have not been used by other media. The 10 classes are used for relevance estimation: given a query image, it is assumed that the user is searching for images from the same class, and therefore the remaining 99 images from the same class are considered relevant and the images from all other classes are considered irrelevant

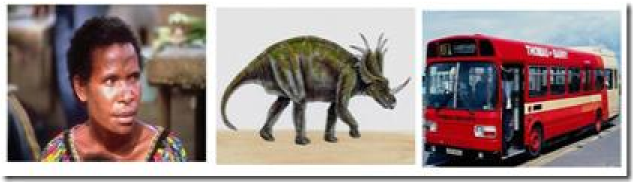


Fig12. sample database images

The CBIR system that is developed is tested using wang database. In each class 100 images are there and only 80 images are trained and 20 are taken into consideration for testing. Two parameters are used for statistical analysis of the system and are precision and recall.

Precision is the ratio of is the ratio of number of relevant images retrived to the total number of images retrieved, whereas recall is the number of relevant images retrieved to the total number of relevant images.

Figure below shows the result of the existing system using human class used for testing.(part of Detected class are shown because of space limitation)

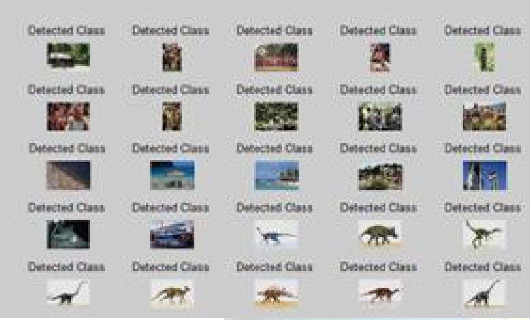


Fig13. Human class used for classification

It is observed from this experiment that the precision value is 0.1739 and recall value is 0.1200.

Table shows the precision and recall values computed for all the classes using the existing technique.

TABLE 1.PRECISION,RECALL FOR EXISTING SYSTEM

Class	Precision	Recall
Humans	0.1739	0.12
Horse	0.1184	0.09
Dinosaur	0.1852	0.05
Rose	0.1121	0.12
Food	0	0
Bus	0.0556	0.03
Mountains	0.1591	0.07
Sceneries	0.1296	0.14
Elephant	0.2	0.08

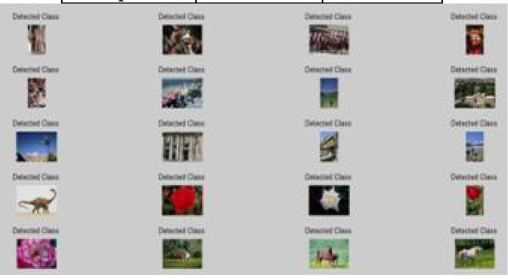


Fig14. Humans class used for proposed technique

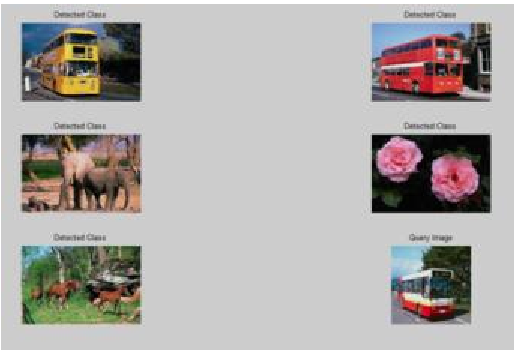


Fig15. Bus class used for proposed technique

TABLE 2.PRECISION,RECALL FOR PROPOSED SYSTEM

Class	Precision	Recall
Humans	0.23	0.05
Horse	0.045	0.01
Dianosur	0	0
Rose	0.218	0.07
Food	0	0
Bus	0.4	0.02
Mountains	0.193	0.06
Scenaries	0.25	0.02
Elephant	0.09	0.01

8.CONCLUSION

In this paper, a novel approach for Content Based Image Retrieval is presented which combines the color and shape features. The proposed algorithm uses color edge detection technique and wavelet-based feature extraction. The proposed approach extracts the edges from Y matrix of YCBCR using Canny edge detection technique and RGB histogram is computed as global statistical descriptor that represents the distribution of colors in an image. The experimental results show that the proposed method performs better even in the case of query image alterations. This system can be extended by increasing the complexity of the wavelets, So that it will reach its ideal conditions, results in exact image.

REFERENCES

[1] Swati Agarwal, A. K. Verma, Preetvanti Singh, “Content Based Image Retrieval Using Discrete Wavelet Transform & Edge Histogram Descriptor”, IEEE International Conference on Information System and Computer Networks (ISCON), pp. 19-23, March 2013.

[2] V. Balamurugan, P. Anandhakumar, "An Integrated Color and Texture Feature based Framework for Content Based Image Retrieval

using 2D Wavelet Transform", IEEE Proceedings of the International Conference on Computing, Communication and Networking, 2008.

[3] Seema Haribhau Jadhav, Shah Aqeel Aehmad, "Content Based Image Retrieval System with Hybrid Feaure set and Recently Retrieved Image Library", (UCSIS) International Journal of Computer Science and Information Security, Vol. 59, No. 5, 2012.

[4] Christo Ananth, A.S.Senthilkani, S.Kamala Gomathy, J.Arockia Renilda, G.Blesslin Jebitha, Sankari @Saranya.S., "Color Image Segmentation using IMOWT with 2D Histogram Grouping", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue. 5, May 2014, pp-1 – 7

[5] S. Soman, M. Ghorpade, V. Sonone, S. Chavan, "Content Based Image Retrieval using Advanced Color and Texture Features", International Conference in Computational Intelligence (ICCIA), 2011.

[6] Elif Albuz, Erturk Kocalar, Ashfaq A. Khokhar, "Scalable Color Image Indexing and Retrieval using Vector Wavelets", IEEE Transactions on Knowledge and Data Engineering, Vol. 13, NO. 5, 2001.

[7] Mark A. Ruzon, Carlo Tomasi, "Color Edge Detection with the Compass Operator", IEEE conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 160-166, 1999.

[8] A. Komali, V. Satish Kumar, K. Ganapathi Babu, A. S. K. Ratnam, "3D Color Feature Extraction in Content-Based Image Retrieval", International Journal of Soft Computing and Engineering (IJSCE) Vol. 2, Issue 3, July 2012.

[9] Tian Yumin, Mei Lixia, "Image Retrieval Based on Multiple Features Using Wavelet", 5th IEEE International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03), 2003.

[10] B. Ramamurthy, K. R. Chandran, "Content Based Image Retrieval for Medical Images Using canny Edge Detection Algorithm", International Journal of Computer Applications (0975-8887), Vol. 17, No. 6, March 20 12.

[11] H. B. Kekre, Tanuja K. Sarode, Sudeep D. Thepade, Shrikant Sanas, "Assorted Color Spaces to Improve the Image Retrieval using VQ Codebooks Generated using LBG and KEVR", International Conference on Tecbnology Systems and Management (ICTSM), Proceedings of International Journal of Computer Applications (IJCA), 2011.

[12] M. R. Zare, R. N. Ainon, W. C. Seng, "Content-based Image Retrieval for Blood Cells", Third Asia International Conference on Modeling and Simulation, 2009.

[13] Gauri Deshpande, Megba Borse, "Image Retrieval with the use of Different Color Spaces and the Texture Feature", International Conference on Software and Computer Applications (IPCSIT), Vol. 9, IACSIT Press, Singapore.

[14] M. H. Saad, H. I. Saleb, H. Konbor, M. Ashour, "Image Retrieval Based on Integration between YCbCc Color Histogram and Texture Feature", International Journal of Computer Tbeory and Engineering, Vol. 3, No. 5, October 2011.

[15] John Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Macbine Intelligence, Vol. PAMI-8, No.6, 1986.