A Structure for Semantic Image Retrieval Using Low Level Attribute and Image Annotation

[1]Ankit Jasoliya, [2]Nidhi Soni
[1]Assistant Professor, [2]Lecturer
[1, 2]Department of Computer Engineering
[1, 2]Parul Polytechnic Institute, Vadodara,India
Email – [1] Ankitjasoliya6006@gmail.com, [2] 231nidhisoni@gmail.com

Abstract: *The information retrieval is one of the most frequent techniques or application in our daily life for data search in different formats and kinds. In this presented work the image based data retrieval technique is investigated and demonstrated. The image retrieval techniques can be divided two major domain first based on text available in image data and second the content inside the image. When the techniques utilize the image contents then such kind of techniques are known as the content based image retrieval. Content based image retrieval technique consumes image features. These features are essential image attributes that are used to identify the image and the advantage of these features are there size. Therefore there are three key features are used namely colour feature, edge feature and their texture. The techniques which include the content based image retrieval, offers a user to search an image based on image query, but these techniques are not supporting the text based query. In addition of that, the technique required some additional techniques to correct the retrieval process such as user feedback, these methods consumes additional time of search. Thus a new technique with hybrid concept is proposed for improving the content based image search. Therefore it includes the concept of image based user query and text based user query support. The proposed technique includes the technique to train the system using the image attribute and text for annotation of image. For identifying the images more accurately the text and image features are used. Finally to retrieve the data (image) using user query (image or text) a KNN algorithm is implemented with it. The implementation of the proposed model is performed using visual studio technology and their performance in terms of time and space complexity is estimated. In addition of that the performance in terms of accuracy and error rate is also provided for demonstrating the relevancy of image search.*

Key Words: *image retrieval, low label attributes extraction, content based image, image annotation, and semantics.*

1. INTRODUCTION:

Basically the image retrieval is a kind of data analysis and retrieval technique. Among them Content based image retrieval technique is an efficient and accurate technique for finding image according to the user query relevance. Therefore the content of images are extracted or distinguished on the basis of the image objects. In order to recover these features for image data retrieval mainly three features namely colour, texture and shape features are utilized. These features are recovers the image data pattern and then the similarity matching techniques or classification techniques are employed to obtain the more nearer images form the image directories. In addition of that for improving the quality of search results in content based image retrieval. There are various other techniques are available that promises to optimize the search results according to the user need such as relevance feedback. by which the user provide the direction of search. Or sometimes the image re-ranking techniques are utilized for re-rank the search results outcomes. In this proposed technique the pseudo relevance feedback technique is utilized for optimizing the search results. The key advantage of the pseudo relevance, which is not consuming additional memory and time for re-rank the search results according to their relevancy. In further discussion the proposed model design concept and their sub component design aspects are discussed in detail. In addition of that for simulation of the proposed methodology simulation architecture is also provided with their description.

2. PROPOSED WORD:

The entire design of the proposed content based image retrieval technique is demonstrated in two major modules. The detailed understanding about these modules is discussed as:

2.1. Training module: in this phase the user train the proposed data model and during training the image, input object tag are incorporated first then after their feature vectors are estimated. The feature vectors are essential properties of the images which is stored in database for future comparisons and results listing.

2.2. Testing: in this phase the user provides query for making search to the data base. In this phase if the input query is in terms of text then the tags are compared with the image contents and results are build and if the query is found in terms of image then the feature vectors of input image is calculated and then after the input image features are compared with the target database.

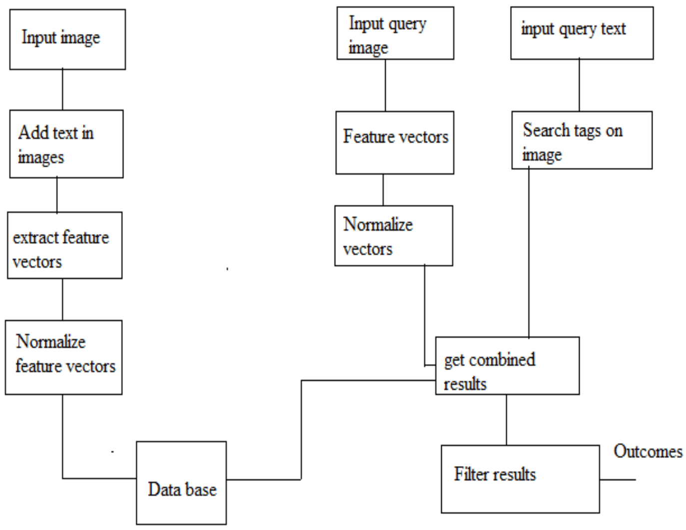

Figure 1 Proposed System

Input image: That is training phase of the system where system accepts user images as input to learn from example.
Add text in images: A small note on image also stored in the data base as annotation of text.
Extract features: In this phase on training system extract different feature vectors and pass these vectors into next phase to normalize them.
Normalize feature vectors: In this phase from multiple values a common value is prepared and stored in data base in place of complete image.
In testing session we use these stored values from the database and consumes to extract the optimum results from the database.
Input query image: User can make query for search results as image from the query image feature vectors are estimated as in training phase.
Feature vectors: Here grid color movement, canny edge detection, local binary patterns are estimated as feature set.
Normalize feature vector: Individual features are contains their own definition and comparison and storage of each features can take more space and time for comparison thus a common value is calculated to store them.
Text input: User can also place text for search image contains.
Search similar tag: The text in images is searched on local database to get images from database.
Combine results: Here the results obtained from text query and image query is listed in same place.
Filter results: To obtain the optimum results from the data omitted as results.
The given figure 1 describes the training process of the proposed data model where first a provision is made to accept the input image data. The image data is further analysed using the region growing algorithm, which extend the image features for improving the image feature quality. In next step the colour features are extracted therefore that need to analyse the image using the grid colour movement analysis. The grid colour movement analysis first convert the entire image into small blocks of images and then the formulas for colour distribution is applied. The computation of the grid colour movement analysis can be described as:

2.3. Grid Colour Moment:

Colour feature is one of the most widely used features in low level feature. Compared with shape feature and texture feature, colour feature shows better stability and is more insensitive to the rotation and zoom of image. Colour not only adds beauty to objects but also more information, which is used as powerful tool in content-based image retrieval. In colour indexing, given a query image, the goal is to retrieve all the images whose colour and texture compositions are similar to those of query image. In colour image retrieval there are various methods, but here we will discuss some prominent methods. The feature vector we will use is called "Grid-based Colour Moment". Here is how to compute this feature vector for a given image: [10]

- Convert the image from RGB for HSV colour space [7]
- Uniformly divide the image into 3x3 blocks
- For each of these nine blocks
- Compute its mean colour (H/S/V)

$$x' = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Where N is the number of pixels within each block, $x_i$is the pixel intensity in H/S/V channels.

- Compute its variance (H/S/V)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - x')^2$$

- Compute its skewness (H/S/V)

$$\gamma = \frac{\frac{1}{n}\sum_{i=1}^{N}(x_i - x')^3}{\left(\frac{1}{n}\sum_{i=1}^{N}(x_i - x')^2\right)^{3/2}}$$

Each block will have 3+3+3=9 features, and thus the entire image will have 9x9=81 features. Before we use SVM to train the classifier, we first need to normalize the 81 features to be within the same range, in order to achieve good numerical behaviour. To do the normalization, for each of the 81 features:

- Compute the mean and standard deviation from the training dataset

$$\mu = \frac{1}{M}\sum_{i=1}^{M} f_i$$

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=0}^{M} (f_i - \mu)^2}$$

Where M is the number of images in the training dataset, and f_i is the feature of the i-th training sample.

- Perform the "whitening" transform for all the data (including both the training data and the testing data), and get the normalized feature value:

$$f'_i = \frac{f_i - \mu}{\sigma}$$

The computed color features of images are preserved in a feature database. And in next step the edge detection technique is applied on image for recovering the edges of the image therefore the canny edge detection technique is applied on image, the canny edge detection technique can be described as:

2.4. Canny Edge Detection:

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several algorithms exists, and this worksheet focuses on a particular one developed by John F. Canny (JFC) in 1986. [5]

The algorithm runs in 5 separate steps:

- Smoothing: Blurring of the image to remove noise.
- Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.
- Non-maximum suppression: Only local maxima should be marked as edges.
- Double thresholding: Potential edges are determined by thresholding.
- Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

2.5. Smoothing:

(a) Original (b) Smoothed

Figure 2. smoothing effect on image

It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the image is first smoothed by applying a Gaussian filter. The kernel of a Gaussian filter with a standard deviation of σ = 1.4. The effect of smoothing the test image with this filter is shown in Figure 2.

2.6. Finding gradients:

The gradient magnitudes (also known as the edge strengths) can then be determined as an Euclidean distance measure by applying the law of Pythagoras.

$$|G| = \sqrt{G_x^2 + G_y^2}$$

It is sometimes simplified by applying Manhattan distance measure to reduce the computational complexity.

$$|G| = |G_x| + |G_y|$$

Gx and Gy are the gradients in the x- and y-directions respectively.
The Euclidean distance measure has been applied to the test image. The computed edge strengths are compared to the smoothed image in Figure (3).

(a) Smoothed (b) Gradient magnitudes

Figure 3. gradient magnitudes of image

That an image of the gradient magnitudes often indicates the edges quite clearly, However, the edges are typically broad and thus doing not indicate exactly where the edges are. To make it possible to determine this, the direction of the edges must be determined and stored as.

$$\theta = arcTan\left(\frac{|G_y|}{|G_x|}\right)$$

2.7. Non-maximum suppression:

The purpose of this step is to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” edges. Basically this is done by preserving all local maxima in the gradient image, and deleting everything else. The algorithm is for each pixel in the gradient image:

- Round the gradient direction θ to nearest 45◦, corresponding to the use of an 8-connected neighbourhood.
- Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction. I.e. if the gradient direction is north (theta = 90◦), compare with the pixels to the north and south.
- If the edge strength of the current pixel is largest; preserve the value of the edge strength. If not, suppress (i.e. remove) the value.

2.8. Double thresholding:

The edge-pixels remaining after the non-maximum suppression step are (still) marked with their strength pixel-by-pixel. Many of these will probably be true edges in the image, but some may be caused by noise or colour variations for instance due to rough surfaces. The simplest way to discern between these would be to use a threshold, so that only edges stronger that a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

2.9. Edge tracking by hysteresis:

Strong edges are interpreted as “certain edges”, and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge (with proper adjustment of the threshold levels). Thus strong edges will (almost) only be due to true edges in the original image. The weak edges can either be due to true edges or noise/colour

INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD ISSN: 2455-0620 Volume - 6, Issue - 7, July – 2020
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value: 86.87 Impact Factor: 6.719
Received Date: 03/07/2020 Acceptance Date: 18/07/2020 Publication Date: 31/07/2020

variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges.

Edge tracking can be implemented by BLOB-analysis (Binary Large OBject). The edge pixels are divided into connected BLOB's using 8-connected neighbourhood. BLOB's containing at least one strong edge pixel are then preserved, while other BLOB's are suppressed. The effect of edge tracking on the test image is shown in Figure.

(a) Double thresholding (b) Edge tracking by hysteresis (c) Final output

Figure 4. Blob analysis

Furthermore the texture analysis of input image is performed thus here the local binary pattern analysis technique is employed on image. The LBP feature analysis technique can be described as:

2.10. Local binary pattern:

Given a pixel in the image, an LBP [21] code is computed by comparing it with its neighbours:

$$LBP_{P,R} = \sum_{P=0}^{P=1} s(g_p - g_e)2^p$$

$$s(x) = \begin{cases} 0 & x \geq 0 \\ 1 & x < 0 \end{cases}$$

Where g_e is the gray value of the central pixel,g_p is the value of its neighbors, P is the total number of involved neighbors and R is the radius of the neighborhood. Suppose the coordinate of is (0, 0), then the coordinates of g_p are

$$\left(Rcos\left(\frac{2\pi p}{P}\right), Psin\left(\frac{2\pi P}{P}\right)\right)$$

The gray values of neighbours that are not in the image grids can be estimated by interpolation. Suppose the image is of size I*J After the LBP pattern of each pixel is identified, a histogram is built to represent the texture image:

$$H(k) = \sum_{i=1}^{I}\sum_{J=1}^{j} f\left(LBP_{p,r}(i,j), k\right), k \in [0,k]$$

$$f(x,y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases}$$

where K is the maximal LBP pattern value. The U value of an LBP pattern is defined as the number of spatial transitions (bitwise 0/1 changes) in that pattern

$$U(LBP_{P,R}) = \left|s(g_{p-1} - g_e) - s(g_0 - g_e)\right| + \sum_{p=1}^{p-1}\left|s(g_p - g_e) - s(g_{p-1} - g_e)\right|$$

The uniform LBP patterns refer to the patterns which have limited transition or discontinuities ($U \leq 2$) in the circular binary presentation. In practice, the mapping from$LBP_{P,R}$ to $LBP_{P,R}^{u2}$ which has P*(P-1)+3 distinct output values, is implemented with a lookup table of 2^P elementsTo achieve rotation invariance, a locally rotation invariant pattern could be defined as:

$$LBP_{P,r}^{riu2} = \begin{cases} \sum_{p=0}^{p-1} s(g_p - g_e) \; if U(LBP_{P,R}) \leq 2 \\ P + 1 \; otherwise \end{cases}$$

The mapping from $LBP_{P,R}$ to $LBP_{P,R}^{u2}$ which has P+2 distinct output values.

Finally the tag on the image is also included and all three features and their tags are saved on the database. This preserved features and tags of the image objects are used for search the data according to the user supplied query. The next section described the test phase of the proposed model.

2.11. Testing:

The testing model of the proposed content based image retrieval model is given using figure 2.5. This model is basically used for extracting the images from data base according to the input query. Therefore a provision is made to provide input for search there are two kinds of input can be applied first by using the image query and second for the text query. User can search the images by the text or image or from both image and text. After user query input the system analyse the input query type and according to the user query if the input is only text then the system search on the tag data base here for comparing the image existing tag and user input text the cosine similarity is utilized. And if the input query is an image than first the query image's low level features are computed and then compared with the feature database. In these conditions for finding the more relevant images from the database the KNN (k nearest neighbour) algorithm is applied. Finally according to the omitted distance from the available images the search results are ranked and reflected as list of images retrieved.

For classifying image the K-nearest neighbour algorithm can be described as:

The K-nearest-neighbour (KNN) algorithm measures the distance between a query scenario and a set of scenarios in the data set. We can compute the distance between two scenarios using some distance function d(x, y), where x, y are scenarios composed of features, such that.

$$X = \{x_1, x_2, x_3, \dots\}$$
$$Y = \{y_1, y_2, y_3, \dots\}$$

Two distance functions are discussed here:

Absolute distance measuring:

$$d_A(x, y) = \sum_{i=1}^{N} |x_i - y_i|$$

Euclidean distance measuring:

$$d_A(x, y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2}$$

Because the distance between two scenarios is dependant of the breaks, it is suggested that resulting distances be scaled such that the arithmetic mean across the dataset is 0 and the standard deviation is 1. This can be accomplished by replacing the scalars with according to the following function:

$$\mathbf{x}' = \frac{x - \bar{x}}{\sigma(x)}$$

Where the un-scaled value is the arithmetic mean of feature across the data set, is its standard deviation, and is the resulting scaled value.

The arithmetic mean is defined as:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

We can then compute the standard deviation as follows:

$$\sigma(x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

KNN can be run in these steps:

1. Store the output values of the M nearest neighbours to query scenario Q in vector r = {r_1…...r_m} by repeating the following loop M times:
   a. Go to the next scenario S_i in the data set, where I is the current iteration within the domain {1……P}
   b. If Q is not set or q < d (q, S_i): q ←d (q, S_i), t ←O_i
   c. Loop until we reach the end of the data set.
   d. Store q into vector c and t into vector r.
2. Calculate the arithmetic mean output across r as follows:

$$\bar{r} = \frac{1}{M} \sum_{i=i}^{M} r_i$$

3. Return r as the output value for the query scenario q

3. RESULTS ANALYSIS:

The implemented model of the proposed image retrieval system is discussed in this chapter. Therefore the performance of both the techniques traditional as given in [1] and the proposed model is compared in this chapter over different performance parameters.

3.1 Memory consumption:

The amount of space in main memory required to execute the algorithm is known as the memory consumption of the algorithm. The performances in terms of memory for both the algorithms are demonstrated using the figure 5.

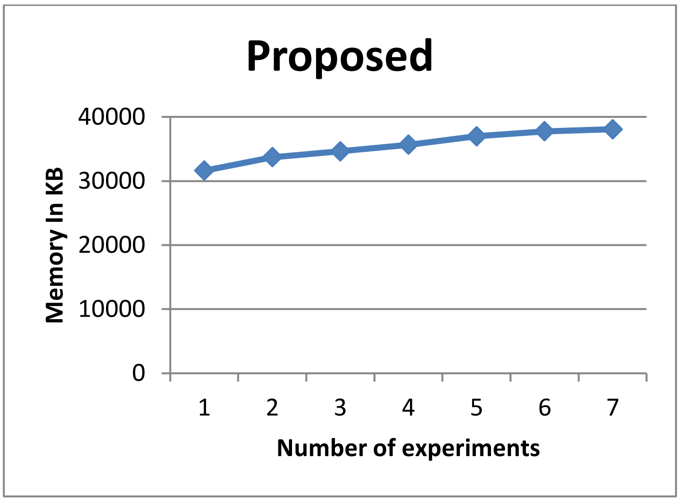

Figure 5. Memory consumption

According to the outcomes as given in figure 3.1 the performance of the algorithm is efficient. But sometimes the memory consumption of the proposed algorithm is higher than 35000KB. In order to demonstrate the performance of the system the X axis contains the different experiments performance with the system and the Y axis shows the memory consumption of the system in terms of KB. The performance of the proposed method is provided using the blue line.

3.2 Training time

The amount of time consumed for train the algorithm is termed here as the training time. The training time of the system is given using figure 6.

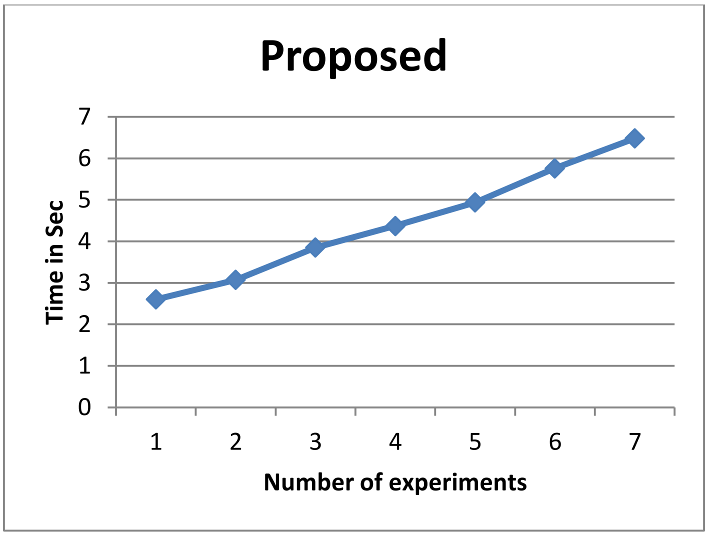

Figure 6. Training time

In this diagram the blue line shows the performance of the proposed system. For representation of the performance the X axis shows the different experiments performed on the system and the Y axis shows the consumed time in second. According to the obtained results the performance of the propose algorithm is consumes less time therefore the performance of the proposed technique is much optimum as compared to traditionally available technique.

3.3 Search time

The search time of the system is termed as the amount of time required to search the images from data base using the input query. The time consumption of the system for the proposed techniques is reported using the figure 7.. In this diagram the X axis shows the number of experiment performed and the Y axis shows the amount of time consumed in terms of seconds. According to the obtained performance the proposed technique consumes less time.

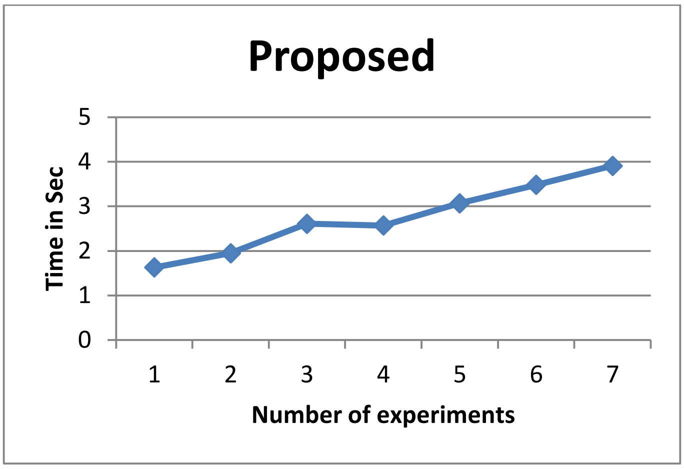

Figure 7. Search time

3.4 Accuracy: In the predictive data models the amount of correctly identified patterns are known as the accuracy of the predictive system. That can be calculated using the following formula.

$$accuracy = \frac{total\ correctly\ classified\ samples}{total\ input\ samples} X100$$

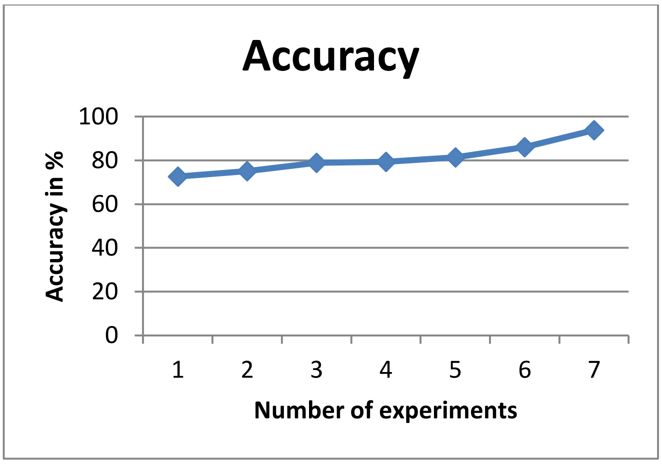

Figure 8. Accuracy

The estimated accuracy of the proposed data model is given using figure 8, in this diagram the X axis shows the number of data samples on which the model performs training and the Y axis simulated the total percentage of samples which are correctly identified. According to the obtained results when the model is trained with less amount of data than the performance of prediction is poorer and as the training samples are increases the performance of the system is enhanced.

3.5 Error rate:

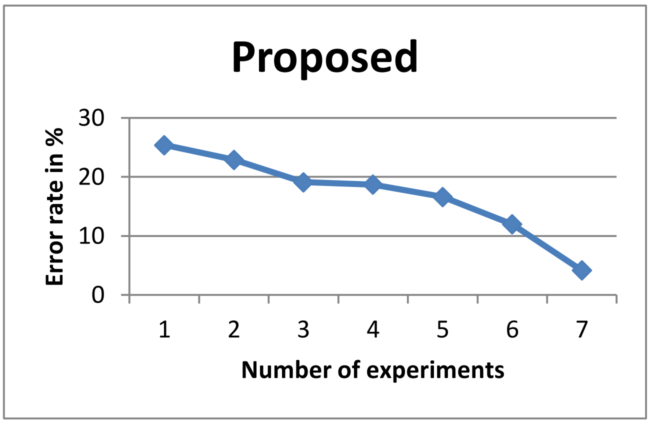

Figure 9. Error rate

The amount of miss-classified samples is known as the error rate of system. That can be estimated using the following formula.

$$error\ rate = \frac{total\ misclassified\ samples}{total\ samples\ as\ input} X100$$

Or

$$error\ rate = accuracy\ value - 100$$

The error rate of the proposed recommender engine is given using figure 3.5 where the X axis contains the training samples as input and the Y axis shows the error rate of system. According to the evaluated results the error rate of the system is decreases as the amount of training samples are increase. Thus the proposed model is adoptable and enhancing their learning capability of system as the samples in data base is increases.

4. CONCLUSION:

Internet is a source of information and data, where different kinds and formats of data are available for use. Internet users can search the specific kind of data from web using the web search engines. The extraction from the data from a given source of data is known as the information retrieval. In this presented work the image retrieval systems are key area of research. During investigation various kinds of image search systems are found some of them are working on the basis of text associated with the images and some of them are works on the basis of image contents. The content based image retrieval systems are much accurate than the text based search systems. But due to annotation concepts that technique can be more appropriate then content based search. In this presented work a hybrid model for image retrieval proposed. That technique usage the concept of image annotation and content based image retrieval. Therefore the technique allows a user to search image by their text query as well as the image query. For design aspects the system first utilizes the feature extraction techniques thus for shape feature the canny edge detection technique, for colour features the grid colour movement analysis and for texture analysis the local binary pattern is investigated. The implementation of the proposed content based image retrieval system is given by the visual studio technology. After implementation of the concept the performance of the technique is also investigated in terms of time and space complexity and the search relevancy is measured in terms of accuracy and error rate. The performance summary of the proposed model is given using table 4.1.

S. No.	Parameters	Proposed method	Traditional system
1	Memory usage	High	Low
2	Training time	Low	High
3	Search time	Low	High
4	Accuracy	High	Low
5	Error rate	High	Low

Table 1 Performance summary

According to the obtained results the performance of the proposed system is optimum as compared to the traditional data model. That produces high accurate image listing during search and also provides the outcomes in less amount of time.

REFERENCES:

1. S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed.,R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
2. J. Breckling, Ed., *TheAnalysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
3. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
4. M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, “Highresolution fiber distributed measurements with coherent OFDR,” in *Proc. ECOC’00*, 2000, paper 11.3.4, p. 109.
5. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.
6. The IEEE website. [Online]. Available: http://www.ieee.org/(2002)
7. M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-
8. *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
9. “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.
10. A. Karnik, “Performance of TCP congestion control with rate feedback:TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute ofScience, Bangalore, India, Jan. 1999.
11. J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Renocongestion avoidance and control,” Univ. of Massachusetts, Amherst,MA, CMPSCI Tech. Rep. 99-02, 1999.
12. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*,IEEE Std. 802.11, 1997.