IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 48-53

Extraction of Image Texture and Color Features: CBIR Techniques

Vaibhav Magar[1], Anjali Rochkari[2], Mahadu.A.Trimukhe [3]
Department of Electronics & Telecommunication Engineering
Terna Engineering College, Nerul

Abstract: *In modern days with the increase of social networking mediums so many digital images are uploaded every day to access this very large image collection New technology introduced that is the Content Based Image Retrieval which is Image mining system. Due to the enormous increase in image database sizes as well as its vast deployment in various applications, the need for CBIR development arose. The need to find a desired image from a large collection is shared by many professional groups, including journalists, design engineers and art historians. Content-based image retrieval (CBIR) is an important research area for manipulating large amount of image databases and archives. CBIR implements retrieval based on the similarities described in terms of extracted features with more goodness. In this paper, dynamic content-based image search and retrieval is conferred called feature extraction method. The perceptible contents of an image such as texture & color feature extractions are availed in CBIR. CBIR requires feature extraction and computation of similarity. The Haar wavelet transform is used for texture feature extraction, and for color feature extraction use color moments. The distance between the query image features and the database images features is computed. Experiment results reflect the importance of the Haar wavelet transform and color moments in the performance of proposed CBIR method.*
Keywords- *CBIR, Haar Wavelet, Color moment, Canberra distance*

I. Introduction

The aim of the paper is to solve the problems of retrieval of images from a large databases. It is not possible to annotate each image in a large database of images. Hence rather than using the metadata, such as keywords, tags & annotations associated with that image we use input as a image, and the output is similar images from the databases. The goal of Content-Based Image Retrieval (CBIR) systems is to operate on collections of images and, in response to visual queries, extract relevant image. The application potential of CBIR for fast and effective image retrieval is enormous, expanding the use of computer technology to a management tool.[4]

While the requirements of image users can vary considerably, it can be useful to characterize image queries into three levels of abstraction: primitive features such as color or shape, logical features such as the identity of objects shown and abstract attributes such as the significance of the scenes depicted. While CBIR systems currently operate effectively only at the lowest of these levels, most users demand higher levels of retrieval. Users needing to retrieve images from a collection come from a variety of domains, including crime prevention, medicine, architecture, fashion and publishing. Remarkably little has yet been published on the way such users search for and use images, though attempts are being made to categorize users' behavior in the hope that this will enable their needs to be better met in the future. Attempts are also going on integrating the search for all kind of images and combining all above mentioned feature vectors for comparison and retrieval so as to achieve the best possible efficiency

II. Texture

An image texture is a set of metrics calculated in image processing designed to quantify the perceived texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image.[5]Image textures can be artificially created or found in natural scenes captured in an image. Image textures are one way that can be used to help in segmentation or classification of images. To analyze an image texture in computer graphics, there are two ways to approach the issue: Structured Approach and Statistical Approach.

2.1. Texture Segmentation

The use of image texture can be used as a description for regions into segments. There are two main types of segmentation based on image texture, region based and boundary based. Though image texture is not a perfect measure for segmentation it is used along with other measures, such as color, that helps solve segmenting in image. Region Based Attempts to group or cluster pixels based on texture properties together. Boundary Based Attempts to group or cluster pixels based on edges between pixels that come from different texture properties In this paper we are going to use region based technique.

III. HAAR WAVELATE Transform

The Haar transform is the simplest of the wavelet transforms. This transform cross-multiplies a function against the Haar wavelet with various shifts and stretches. The Haar transform is one of the oldest transform functions, proposed in 1910 by the Hungarian mathematician Alfréd Haar. It is found effective in applications such as signal and image compression in electrical and computer engineering as it provides a simple and computationally efficient approach for analysing the local aspects of a signal.The Haar transform is derived from the Haar matrix.[3] An example of a 4x4 Haar transformation matrix is shown below.

$$H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

The Haar transform can be thought of as a sampling process in which rows of the transformation matrix act as samples of finer and finer resolution.The Haar transform has the following properties

1. No need for multiplications. It requires only additions and there are many elements with zero value in the Haar matrix, so the computation time is short. It is faster than Walsh transform, whose matrix is composed of +1 and −1.
2. Input and output length are the same. However, the length should be a power of 2, i.e. $N = 2^k, k\Sigma N$
3. It can be used to analyze the localized feature of signals. Due to the orthogonal property of the Haar function, the frequency components of input signal can be analyzed.

3.1Color Moment

We use color moments for color feature extraction. The basis of color moments lies in the assumption that the distribution of color in an image can be interpreted as a probability distribution.[4] Probability distributions are characterized by a number of unique moments (e.g., normal distributions are differentiated by their mean and variance). Computing color moments is done in the same way as computing moments of a probability distribution.

Mean

The first color moment can be interpreted as the average color in the image and can be calculated by using the following formula:

$$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij}$$

where: N = number of pixels in the image, p_{ij}= value of the j-th pixel of the image at the i-th color channel.

Standard Deviation

The second color moment is the standard deviation, which is obtained by taking the square root of the variance of the color distribution.

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E)^2))}$$

Where E_i = mean value, or first color moment, for the i-th color channel of the image

Skewness

The third color moment is the skewness.

$$S_i = 3\sqrt{(\frac{1}{N}\sum(P_{ij} - E)^3)}$$

Kurtosis

The fourth color moment, and, similarly to skewness is kurtosis, it provides information about the shape of the color distribution. More specifically, kurtosis is a measure of how flat or tall the distribution is in comparison to normal distribution [17].

$$K = \frac{E(x-\mu)^4}{\sigma^4}$$

where μ is the mean of x, σ is the standard deviation of x, and $E(t)$ represents the expected value of the quantity t. Color moments can be computed for any color model. Three color moments are computed per channel (e.g. 9 moments if the color model is RGB and 12 moments if the color model is CMYK). Computing color moments is done in the same way as computing moments of a probability distribution.We get nine numbers—twelve moments for each color channel as color features for each of the image.

3.2 Similarity Measures and Distance

The similarity between two images is computed by calculating the distance between feature representation of the query image and feature representation of the image in the dataset. We have used Canberra distance and city block distance for distance calculation of the feature vectors.[6] If the distance between feature representation of the query image and feature representation of the database image is small, then it is considered similar.

3.2.1. Canberra distance

The Canberra distance is a numerical measure of the distance between pairs of points in a vector space.[7] The Canberra distance *d* between vectors **p** and **q** in an *n*-dimensional real vector space is given as follows:

$$d(p,q)\sum_{i=1}^{n} \frac{|p_{i-}q_i|}{|p_i| + |qi|}$$ Where $P = (p1, p2, ...p_n) and q = (q1, q2, ...q_n)$ are vectors.

3.2.2 City Block Distance: A function of the similarity between two image distributions is defined as the sum of the weighted differences between the moments of the two distributions. Formally this is,

$$d_{mom}(H,I) = \sum_{i=1}^{r} w_{il} = \left|E^1{}_i - E^2{}_i\right| + w_{i2}\left|\sigma^1{}_i - \sigma^2{}_i\right| + w_{i3}\left|s^1{}_i - s^2{}_i\right|$$

Where

(H,I) : are the two image distribution being compared i : is the current channel index r : is the number of channels

E_i^1, E_i^2 : are the first moments of two image distributions

3.2.3 Combining the Features

We will work with two feature extraction methods, Haar wavelets for texture feature extraction, and color moments for color feature extraction. Using only a single feature for image retrieval may be inefficient. To produce efficient results, we will combine the two features by adjusting appropriate weights. Similarity between two images is computed by calculating the distance between the feature vectors of the two images. In our paper, we calculate two distances from two feature vectors for each image.[4] Then the final distance is calculated by combining weights with those two distances. The final distance between the query image and the image in the database is calculated as follows:

$$d = d_1 * w_1 + d_2 * w_2$$

Where:

d_1=calculated distance using Haar wavelet features

w_1=weight for Haar wavelet features

d_2=calculated distance using Color features

w_2=weight for Color features

3.4 CBIR

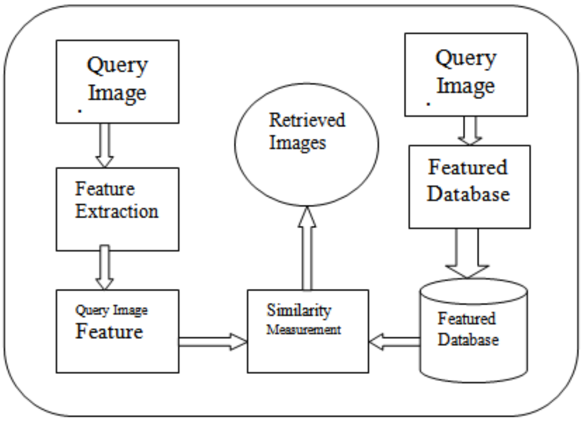

Figure 1. A CBIR system block diagram

CBIR systems work in the following way.[1] In CBIR system an user gives an input as a query image from which a feature vector is extracted, then feature vector is also computed for each image in the database and the set of all feature vectors is organized as a database index. When similar images are searched with a query image, a feature vector is extracted from the query image and is matched against the feature vectors in the database. Differences between the various systems lie in the features they extract and the algorithms used to extract those features.[11] Whose difference distance is lesser that image is displayed as the relevant output images from the databases as the results.

IV. Result

The images in the database are classified into 18 classes and each class contains 30 images. Retrieved image is considered matched if it is in the same category as a query image. The three methods which are implemented are Texture Feature Extraction Method, Color Feature Extraction Method, and Combination of both (Texture + Color Feature).

The performance of a CBIR system can be measured in terms of its precision and recall. Precision measures the retrieval accuracy; it is the ratio between the number of relevant images retrieved and the total number of images retrieved. Recall measures the ability to retrieve all relevant images in the database. It is the ratio between the number of relevant images retrieved and all of the relevant images in the database

Database		Texture Feature		Color Feature		(Texture + Color Feature)	
		Precision(%)	Recall(%)	Precision (%)	Recall(%)	Precision (%)	Recall (%)
1)	Bungalows	78.97%	53.33%	51.72%	53.33%	55.22%	56.41%
2)	Bridges	93.75%	76.67%	62.50%	66.67%	75%	70%
3)	Lakes	46.87%	43.33%	65.21%	53.33%	75%	60%
4)	Forts	41.11%	40.04%	62.50%	56.67%	75%	63%
5)	Dinosaurs	83.31%	76.67%	98%	93.33%	98%	93.33%
6)	Roses	93%	66.67%	60.12%	60.54%	68.18%	63.33%
7)	Buses	83.35%	56.67%	78.98%	53.33%	93.75%	53.37%

Table 1: Precision and Recall Table

We extracted the Texture Features and Color Features separately, and then we experimented with the CBIR system by using a combination of both the features. Table 1 show the Precision and Recall values of the Texture feature, Color feature and combination of both the features of database images of different classes.

4.1 Texture Feature Result

The Figure.2. Shows the result of Texture Feature Extraction method. We have taken a query image number 255 from the database and system retrieves the most similar images from the database.

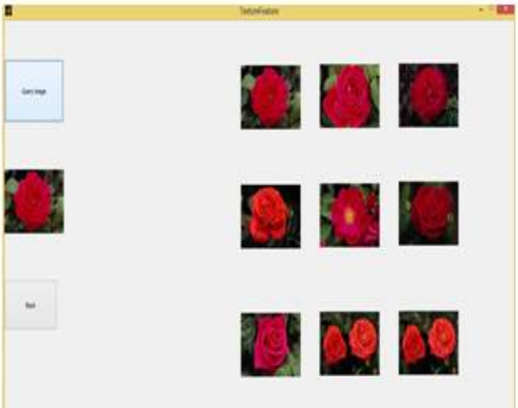

Figure 2. Texture Feature Extraction Result

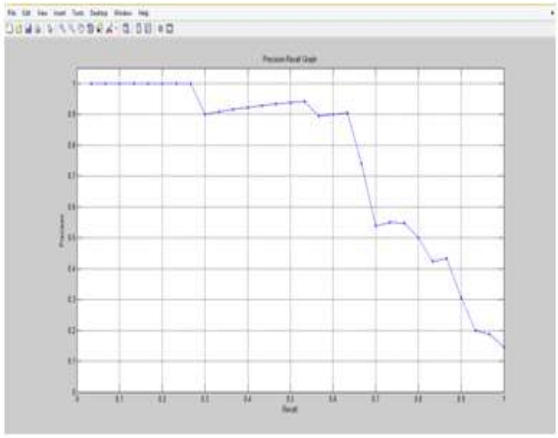

Figure 3. Precision Vs Recall Graph for Texture Feature Extraction .

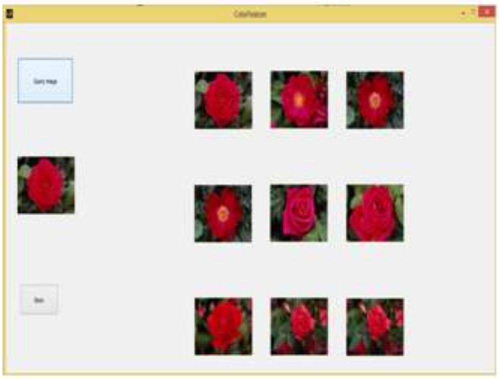

Figure 04. Color Feature Extraction Result.

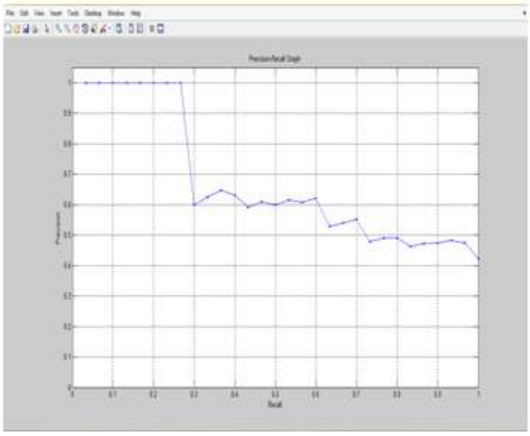

Figure 05. Precision Vs Recall Graph for Color Feature Extraction

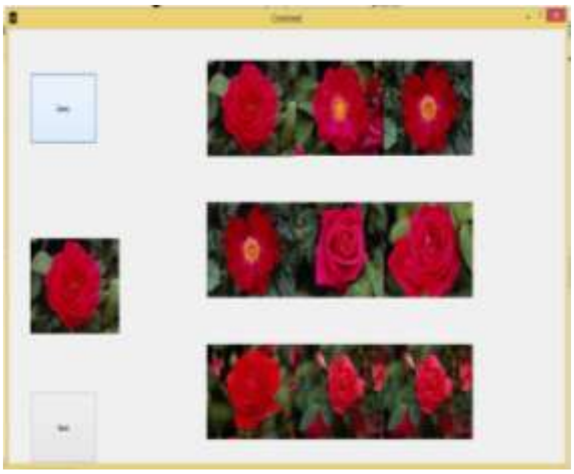

Figure 06. Combined (Texture + Color) Feature Extraction Result

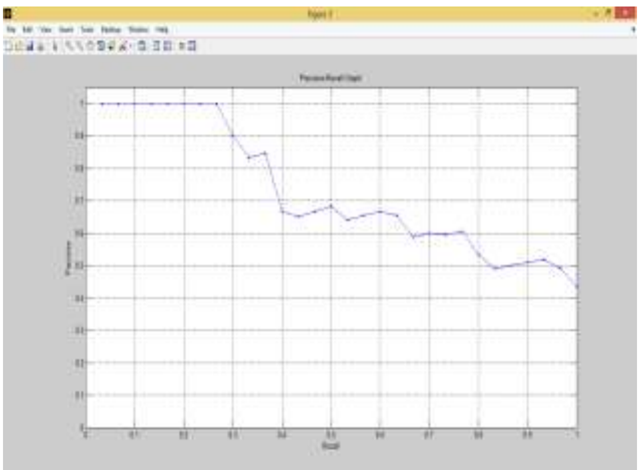

Figure 07. Precision Vs Recall Graph for Combined (Texture + Color) Feature Extraction

V. Conclusion

In this paper, we have proposed an efficient CBIR method based on the Texture feature extraction using Haar wavelet transform and Color feature extraction using color moments. We have included a database of 540 images containing 18 classes each of 30 images. Performance is measured using precision and recall measures for comparison. In distance calculation, Canberra distance gives better result for texture feature extraction and City block distance is efficient for extracting color feature. In most of the retrievals feature extraction using Haar wavelet transform gives better result than Color moment used for Color feature extraction. Hence Haar wavelet transform gives better result than color moment. Combination of both texture and color feature gives better result than all other methods.

References

[1]. Smart Computing Review, vol. 3, no. 3, June 2013.
[2]. Shu-Ching Chen, Senior Member, IEEE, Stuart H. Rubin, Senior Member, IEEE, Mei-Ling Shyu, Senior Member, IEEE, and Chengcui Zhang, Member, IEEE "A Dynamic User Concept Pattern Learning Framework for Content-Based Image Retrieval"- Applications and Reviews,vol.36.november 2006.
[3]. Dr.H.B.Kekre, Sudeep D. Thepade and Akshay Maloo "Query by Image Content using Color-Texture Features Extracted from Haar Wavelet Pyramid" IJCA Special Issue on "Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications" CASCT, 2010.
[4]. Simardeep Kaur and Dr. Vijay Kumar Banga "Content Based Image Retrieval: Survey and Comparison between RGB and HSV model" International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue4- April 2013.
[5]. Henning Muller, Wolfgang Muller , David McG. Squire, St_ephane Marchand-Maillet and Thierry Pun "Performance Evaluation in Content based image retrieval: overview and proposal" Computer vision group university of Geneva.
[6]. Amandeep Khokher and Rajneesh Talwar "Content-based Image Retrieval: Feature Extraction Techniques and Applications" International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012) Proceedings published in International Journal of Computer Applications® (IJCA)
A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall.Learning distance functions using equivalence relations. In ICML, pages 11–18, 2003.
[7]. H. Bay, T. Tuytelaars, and L. J. V. Gool. Surf: Speeded up robust features. In ECCV (1), pages 404–417, 2006.
[8]. B. C. Becker and E. G. Ortiz. Evaluating open-universe face identification on the web. In CVPR Workshops, pages 904–911, 2013.
[9]. Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. CoRR, abs/1206.5538, 2012.
[10]. H. Chang and D.-Y. Yeung. Kernel-based distance metric learning for content-based image retrieval. Image and Vision Computing, 25(5):695–703, 2007.
[11]. G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scaleonline learning of image similarity through ranking. Journalof Machine Learning Research, 11:1109–1135, 2010.
[12]. D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In NIPS, pages 2852–2860, 2012.
[13]. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. Journal of Machine Learning Research, 7:551–585, 2006.
[14]. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker,K. Yang, and A. Y. Ng. Large scale distributed deep networks. In NIPS, pages 1232–1240, 2012.
[15]. L. Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions onSignal and Information Processing, 3:e2, 2014.
[16]. C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. IEEE Trans. Pattern Anal. Mach. Intell., 24(9):1281–1285, 2002.
[17]. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. CoRR,abs/1310.1531, 2013.
[18]. R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Richfeature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524, 2013.
[19]. M. Guillaumin, J. J. Verbeek, and C. Schmid. Is that you?metric learning approaches for face identification. In ICCV, pages 498–505, 2009.
[20]. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed,N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speechrecognition: The shared views of four research groups.Signal Processing Magazine, IEEE, 29(6):82–97, 2012.
[21]. G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learningalgorithm for deep belief for deep belief nets. Neural Computation, 18(7):1527–1554, 2006.