Content Based Image Retrieval System (CBIR) using RGB and HSV Histograms

Hm Adnan Zami[1], Fayezah Anjum[2], Raihanur Rahman[3]

[1,2,3,]Department of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh.

[1]hm.zami@northsouth.edu,
[2]fayezah.anjum@northsouth.edu,
[3]raihanur.rahman21@northsouth.edu

1. Introduction

Text-based image retrieval (TBIR) and content-based image retrieval (CBIR) are the two main mechanisms for image retrieval systems. Searching digital images from a massive database is a complex process. To mitigate this issue, CBIR has turned into an important research area.

Content-based image retrieval (CBIR) is a technique that is used to retrieve visually similar images from a large database of images, for a query image given as input. CBIR is also acknowledged as a query by image content (QBIC). The algorithmic way of searching images from a large pool of data has been made easier with the accessible data available for the images.

CBIR focuses on the content of the images for analyzing purposes. These contents are textures, shapes, and colors from the image itself. Descriptions, tags, or keywords associated with the image can be used to search images instead of using the color, shapes, and textures to find them. However, using data like tags or descriptions to search images is extremely time-consuming. Using CBIR, image retrieval can be measured by precision and recall. Later the measurement can be arranged according to the comparison.

One main problem faced by the CBIR system is the semantic gap problem, which is difficulty in obtaining unique correspondence between the feature vector and image label. This happens because the computer cannot understand the semantic information in the image, and the image is usually categorized using low-level features such as color, shape, and texture. These results in different semantic labels can have a closely similar feature vector value, which is undesirable. Some examples of CBIR are: FIRE [1], AMORE [2], STRSM[3] etc.

The aim of our project is to implement content-based image retrieval (CBIR) by using a histogram-based approach. Color is an essential feature for an image retrieval system. In this paper, the image retrieval system is based on the color histogram of RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) color spaces.

2. Methodology

Color histogram is the representation of the frequency distribution of color bins in an image. In this paper, the HSV color histogram is chosen as the feature vector for the images. To do this, the RGB information is extracted from the images and has been converted into HSV format using the formula given below [4][5].

$$H = \begin{cases} \frac{G-B}{V-\min\{R,G,B\}} \cdot 60°, & \text{if } V = R \text{ and } G \geq B; \\ \left(\frac{B-R}{V-\min\{R,G,B\}} + 2\right) \cdot 60°, & \text{if } G = V; \\ \left(\frac{R-G}{V-\min\{R,G,B\}} + 4\right) \cdot 60°, & \text{if } B = V; \\ \left(\frac{R-B}{V-\min\{R,G,B\}} + 5\right) \cdot 60°, & \text{if } V = R \text{ and } G < B \end{cases}$$

$$S = \frac{V - \min\{R, G, B\}}{V} \quad S \in [0, 1]$$

$$V = \max\{R, G, B\} \quad V \in [0, 255] \qquad (1)$$

After that, the color histogram was constructed by applying a quantization process to reduce the number of bins by putting similar color pixels into the same bins.

CBIR system retrieves a collection of images which have similarities with the query image. There are different types of similarity measurements that are used to determine the similarity between the query image and the images in the dataset. For this project, Euclidean distance has been implemented, a distance metric that gives a numerical measure to compare the degree of similarity between two images. The equation for measuring the Euclidean distance is given below.

$$D = \sqrt{\sum_{i=1}^{k} (hist1 - hist2)^2} \qquad (2)$$

The images that belong to a similar group will be having less Euclidean distance between the feature vectors, and thus the images of the same category can be retrieved using the CBIR system. Euclidean distance uses two feature vectors of the uniform extent and provides a numerical value that tells the degree of similarity between two images. The smaller the Euclidean distance, the more those groups of images will have a similarity.

The flowcharts outline the significant steps of the project. It has been implemented in Matlab, and the dataset used is known as Wang Dataset, which consists of 1000 images of 10 different classes.

Before loading the query image, the data preprocessing has been done, as shown in figure 1. The annotation is done by labeling each image with its corresponding class. Then feature vectors, both in RGB and HSV, are generated and saved in the database.

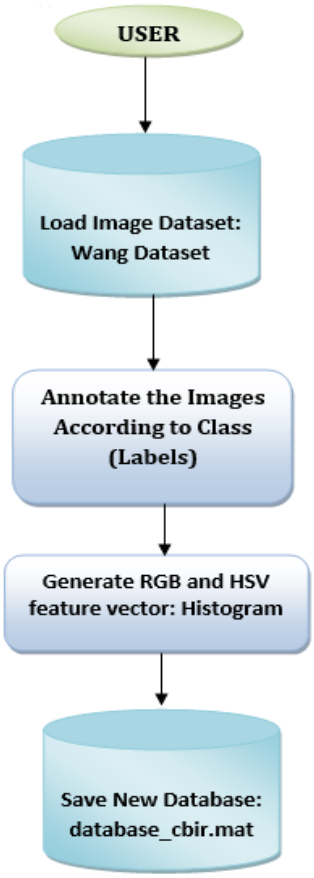


Figure 1: Steps for Data Preprocessing

The flowchart shown in figure 2 gives an overview of how the system works after the data preprocessing. A user can load the query image in either of the color space, RGB or HSV. The input images need to be changed from RGB to HSV for the HSV histogram as a feature vector. Once the RGB/HSV feature vectors of the query images are obtained, the feature vector of the query image is compared against the feature vector of all other images in the database.

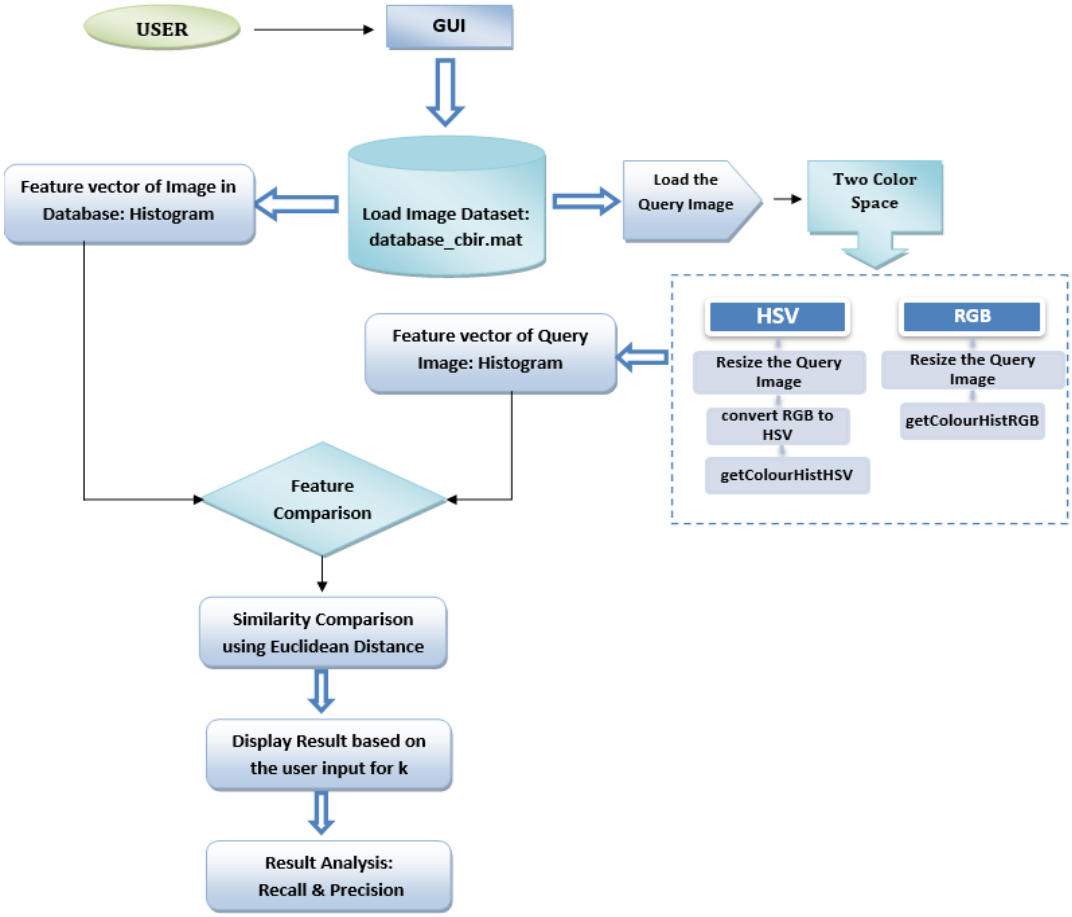


Figure 2: Flowchart of the proposed CBIR System

The detailed procedure of how the project has been implemented in Matlab, along with the user interface, has been discussed in the following Experiment section.

3. Experiment

The CBIR system created for this experiment was done using MATLAB. Initially, the WANG dataset was preprocessed to create a new database that was annotated with respect to the image classes. Then the feature vectors were calculated and stored in the new database. A GUI was designed to load the database and take query images as user input. Once the user selects the number of images to retrieve, the system extracts the feature vector of the query image and compares it with the feature vectors in the database to retrieve similar images.

3.1 Annotate Dataset

The initial step of the project was to create a database with the Wang dataset. This database will store the name of images and its respective classes. Wang's dataset has a total of 1000 images with 10 classes and 100 images per class. To annotate this, we assigned a label for each image according to Table 4.1.

File name	Label	Class
0-99.jpg	1	Africa
100-199.jpg	2	Beach
200-299.jpg	3	Building
300-399.jpg	4	Bus
400-499.jpg	5	Dinosaur
500-599.jpg	6	Elephant
600-699.jpg	7	Flower
700-799.jpg	8	Horse
800-899.jpg	9	Mountain
900-999.jpg	10	Food

Table 4.**1**: Class Labels



Figure 4.1: Image 1 from dataset

3.2 Extract RGB and HSV Feature Vectors

The process of extracting RGB and HSV feature vectors is similar except that HSV feature vectors require an extra step which is converting image RGB color space to HSV. The functions used for retrieving RGB and HSV feature vectors were '*getColourHistHSV.m'* and '*getColourHistHSV.m'.* First, images were resized and the double value was taken. Then each color space was selected and quantization was performed using the function '*doQuanMatrix.m'*. This was necessary to display the color spectrum within a range that would be suitable for system performance as well as results.

After a few trials, the min-max index and levels or bin counts were selected for RGB and HSV which are shown in Table 4.2.

	Hue (H)	Saturation (S)	Value (V)		Red(R)	Green (G)	Blue(V)
Min index	0	0	0		0	0	0
Max index	1.01	255	255		255	255	255
Levels	8	8	4		8	8	4

Table 4.2: HSV and RGB Quantization Values

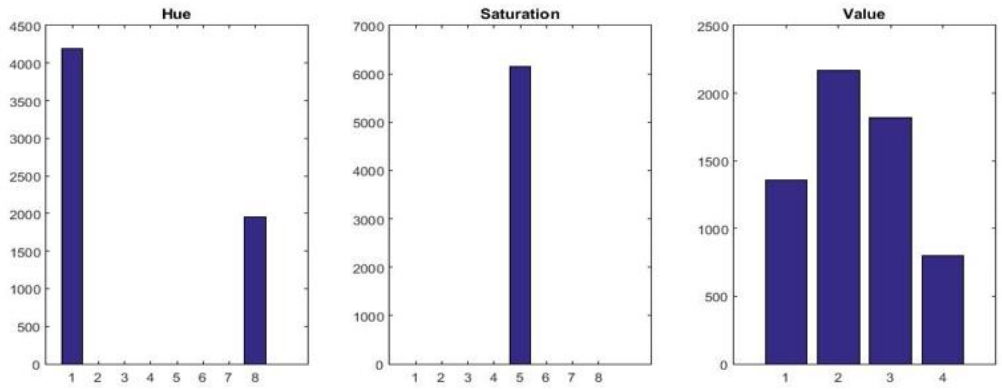


Figure 4.2: H, S and V histogram of Image 1

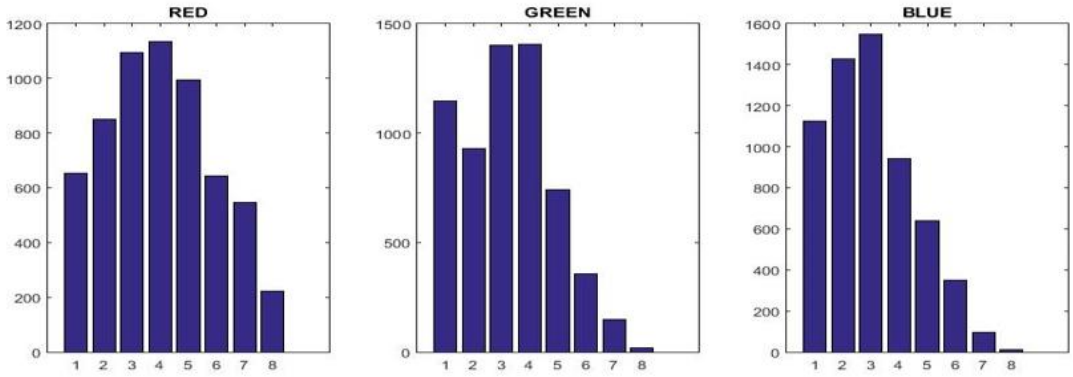
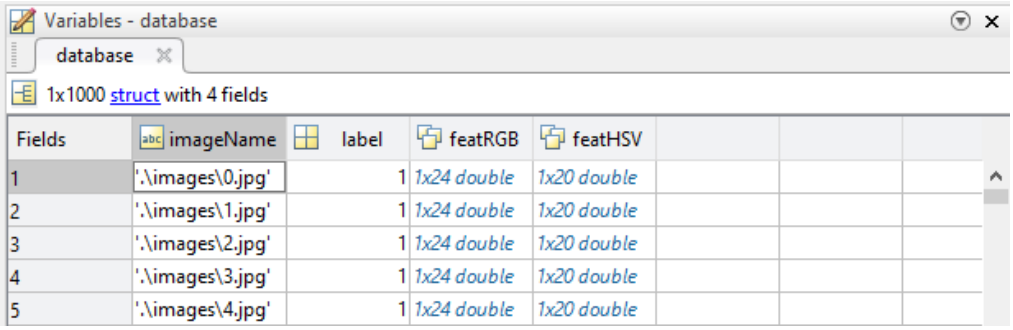


Figure 4.3: R, G and B histograms of Image 1

After performing quantization for each plane, the histograms were retrieved for each channel. The three channels were then concatenated and normalized to produce the final histogram. This histogram was then saved in the database along with the image name and class label.

The final database has 4 fields which are shown in Figure 4.4.



Fields	imageName	label	featRGB	featHSV
1	'.\images\0.jpg'	1	1x24 double	1x20 double
2	'.\images\1.jpg'	1	1x24 double	1x20 double
3	'.\images\2.jpg'	1	1x24 double	1x20 double
4	'.\images\3.jpg'	1	1x24 double	1x20 double
5	'.\images\4.jpg'	1	1x24 double	1x20 double

Figure 4.4 Database Fields

3.3 Get Query Image Feature Vector

To retrieve the image, the same process if followed to get the feature vector of the query image. This time however, the function '*get_image_HSV.m'* and '*get_image_RGB.m'* is called for HSV and RGB query respectively. User will also need to a specify the number of images to retrieve which is stored in variable 'k'. In Figure 4.6, the GUI shows Image 666 as the query image loaded using the HSV method.

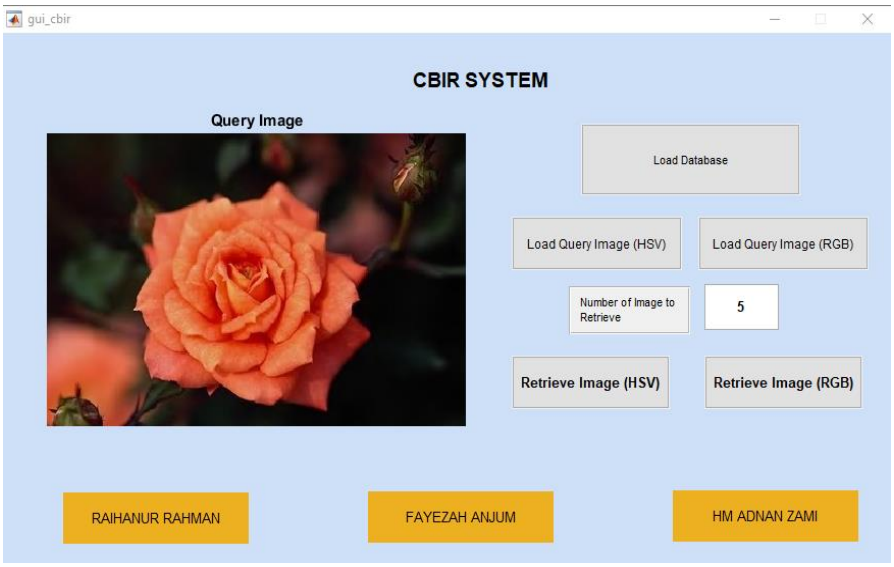


Figure 4.5: Query Image 666 using Load Query Image (HSV)

3.4 Retrieve Similar Images

The histogram of the query image and 'k' is taken as inputs for these functions. In this function, the database is first loaded and the Euclidean distance is calculated for the query image and database image histograms. Then the number of images defined by 'k' is displayed. The precision is also calculated and displayed in the command window. Since the first image is exactly the same as the query image, it is ignored when calculating the precision and also ignored when retrieving 'k' number of images. For comparing images that are not in this dataset, the image needs to be saved in the '*images*' folder before being loaded into the GUI as the query image.

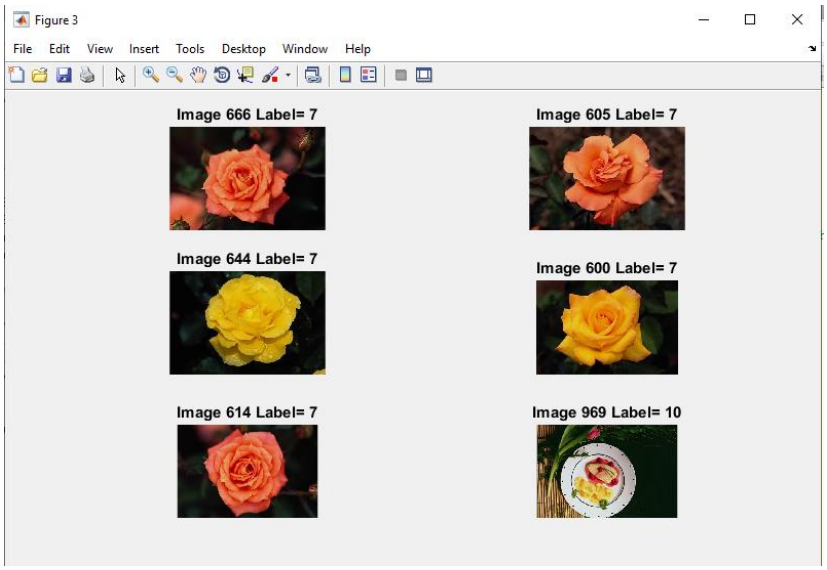


Figure 4.6: 5 similar images retrieved

4. Results and Discussions

In this project, a test set of 10 images were used for both RGB and HSV. These images were randomly selected from each class. A total of 10 images were retrieved from the database after which precision and recall were calculated to test the system's performance. Two evaluation metrics that were used are recall and precision. Precision determines how many images were correctly retrieved out of the total number of images retrieved. On the other hand, recall evaluates the systems performance at retrieving the number of correct images out of the total number of images in the database. The formulas for these metrics are provided below.

. Precision = TruePositives / Total number of images retrieved by the system

Recall = TruePositives / Number of relevant images in database

Our Results

- HSV Precision and Recall (k=10)

The following table shows the Precision and by extension Recall avg. of 5 (query)images from each class set using the HSV search protocol.

Ex: (Africa, im1) is an Image from Africa set and returns a precision of 40% when k=10.

Class	Im 1	Im 2	Im 3	Im 4	Im 5	Precision Avg		Recall Avg
Africa	40%	10%	30%	30%	40%	30%		3%
Beach	30%	60%	50%	50%	70%	52%		5.2%
Building	30%	30%	40%	30%	40%	34%		3.4%
Bus	50%	50%	40%	70%	60%	54%		5.4%
Dinosaur	100%	100%	100%	100%	100%	100%		10%
Elephant	70%	80%	30%	30%	40%	50%		5%
Flower	70%	80%	80%	90%	90%	82%		8.2%
Horse	60%	70%	80%	80%	100%	78%		7.8%
Mountain	70%	30%	30%	70%	30%	46%		4.6%
Food	20%	60%	30%	30%	20%	32%		3.2%

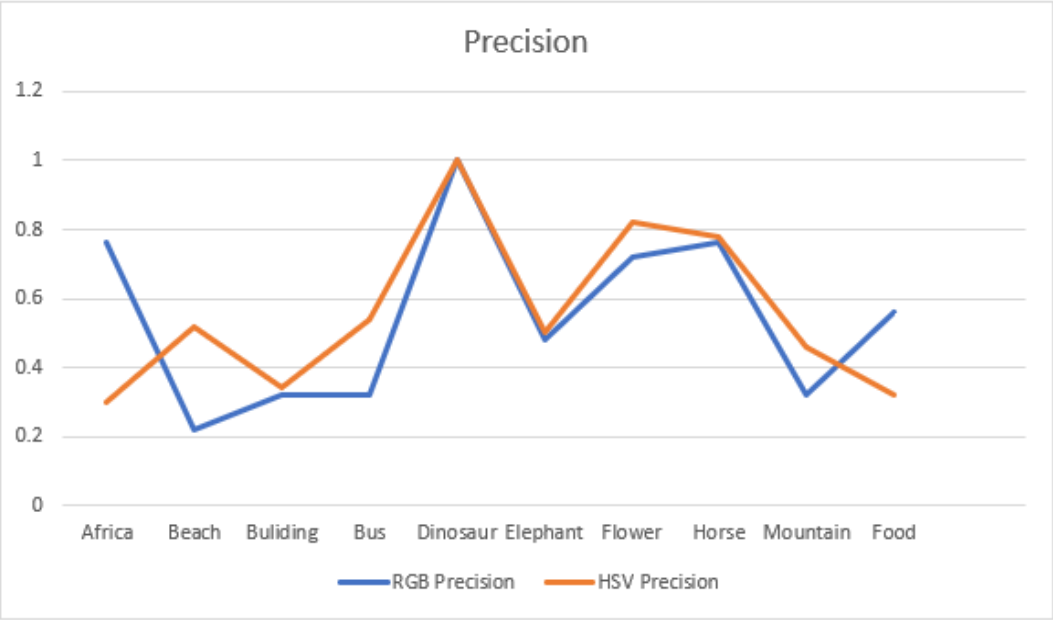
- RGB Precision and Recall (k=10)

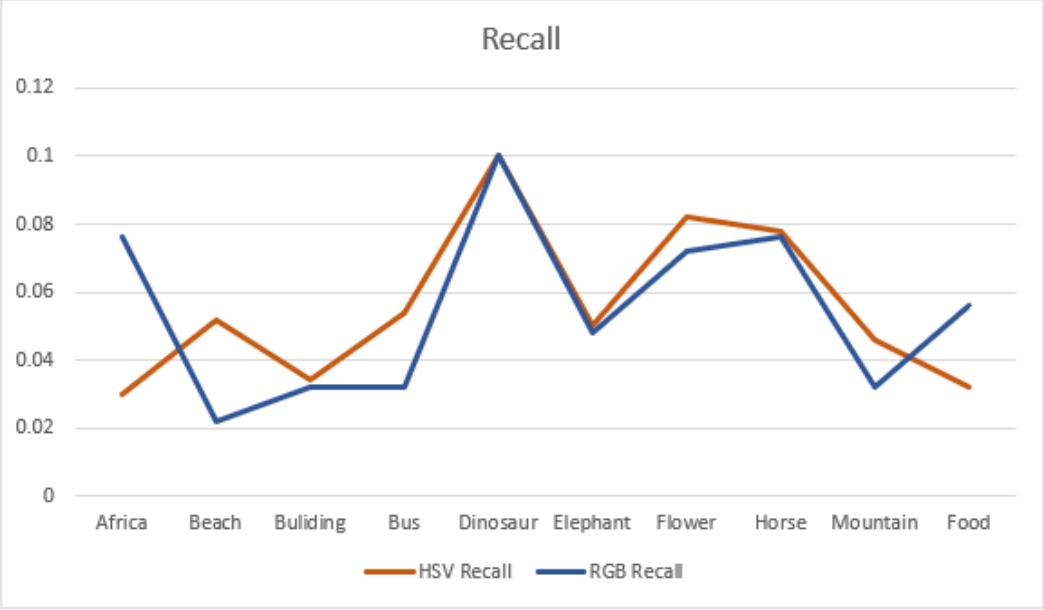
The below table shows the Precision and by extension Recall avg. of 5 (query)images from each class set using the RGB search protocol.

Ex: (Africa, im1) is an Image from Africa set and returns a precision of 80% when k=10.

Class	Im 1	Im 2	Im 3	Im 4	Im 5	Precision Avg		Recall Avg
Africa	80%	70%	80%	50%	100%	76%		7.6%
Beach	20%	10%	10%	50%	20%	22%		2.2%
Building	60%	60%	10%	20%	10%	32%		3.2%
Bus	70%	20%	30%	10%	30%	32%		3.2%
Dinosaur	100%	100%	100%	100%	100%	100%		10%
Elephant	100%	20%	30%	50%	40%	48%		4.8%
Flower	60%	90%	100%	80%	30%	72%		7.2%
Horse	90%	90%	20%	100%	80%	76%		7.6%
Mountain	20%	30%	50%	40%	20%	32%		3.2%
Food	90%	10%	60%	100%	20%	56%		5.6%

Here we can see for the majority of classes, the HSV returns more accurate (higher precision) results compared to RGB. The ratio of Recall Comparison is the same as that of the Precision comparison graph.





After finding our results, we compared it with another paper that performed HSV based CBIR on Wang dataset. Their result further proves the performance of HSV is far greater than RGB when it comes to CBIR system.

The main difference between the related work and ours are 3 things:

i) They calculated the overall average precision as opposed to us calculating it for each class

ii) They didn't use RGB vectors and hence did not compare the difference between the 2.

iii) Their plot shows the individual components of H, S, V , combined HSV.

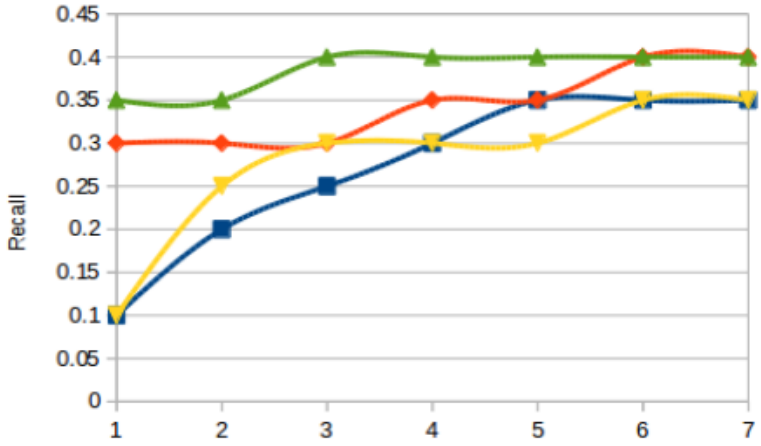


Figure 5.1: Recall Graph for HSV color components

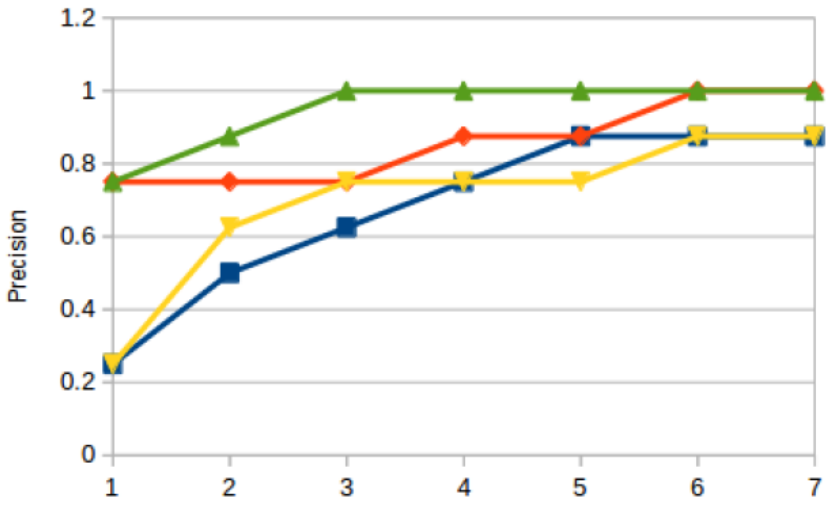


Figure 5.2: Precision Graph for HSV color components

5. Conclusion

CBIR system is an effective way to retrieve images of the same category as it does not involve human labor to do the labeling, which is time-consuming and subjective. It is also more feasible than manual annotation when the involving database is vast.

This paper is about content-based image retrieval (CBIR) by using two histogram-based approaches: RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value). The color histograms are used for the feature vectors of the images. The entire project has been implemented in Matlab, and the Wang Dataset, consisting of 1000 images of 10 different classes, has been used. Depending on the number of images the user wants to retrieve, i.e., the value of k, the result has been displayed. Lastly, the result obtained has been used to calculate the recall and precision values for accuracy analysis. The results of our simulation showed that the HSV featured vector retrieval is more superior to the RBG.

However, the existing techniques show good results only on certain classes but not on the overall dataset. There is a scope of improvement for future work. Instead of only a histogram-based approach, other methods of feature extraction for CBIR can be implemented and compared with the existing system. Expanding the dataset and incorporating machine learning models can also be possible future work.

Reference

[1] T. Deselaers, D. Keysers, and H. Ney, "FIRE -- Flexible Image Retrieval Engine: ImageCLEF 2004 Evaluation", CLEF Workshop 2004, vol. 3491, Bath, UK, Springer, pp. 688-698, 15/09/2004, 2005.

[2] S. Mukherjea, K. Hirata, and Y. Hara, “Amore: a world wide web image retrieval engine”, World Wide Web, 2(3) :115–132, 1999.

[3] Roxana Rosu; Marc Donias; Lionel Bombrun; Salem Said; Olivier Regniers; Jean-Pierre Da Costa “Structure Tensor Riemannian Statistical Models for CBIR and Classification of Remote Sensing Images”, 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) Year: 2016

[4] Z. Wang, A. C. Bovik, and H. R. Sheik, Simoncelli E.P., "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol.13, no.4pp. 600- 612, April 2004.

[5] Z. Wang, and A. C. Bovik, "Mean Squared Error: Love It or Leave It?”, IEEE Signal Processing Magazine [98], January 2009.

[6] Z. Wang, A. Bovik, H.R. Sheikh, and E. P. Simoncelli: 'Image quality assessment: from error visibility to structural similarity', IEEE Transactions on Image Processing, 2004, 13, (4), pp. 600- 612.

[7] Binatha C, 2016, Hue Saturation Value (HSV) Color Space for Content based Image Retrieval, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 05, Issue 04 (April 2016), http://dx.doi.org/10.17577/IJERTV5IS040251