



## Color Image Matching Techniques Using Various Features

Vinky \*

*Student Masters of Technology  
Department of CSE  
Sri Guru Granth Sahib World University  
Fatehgarh Sahib, Punjab, India*

Rajneet kaur

*Assistant Professor & Head of Department of  
Computer Science and Engineering  
Sri Guru Granth Sahib World University  
Fatehgarh Sahib, Punjab, India*

**Abstract---** Comparing images, or image patches, is the desired goal in several computer vision applications. This paper shows a study of various image matching and retrieval criteria for colored images using various features.

**Keywords---** Color Image matching and retrieval, RGB contrast, Maximum Frequency, Minimum Frequency, Shape, Texture, Relativity, Sorting, Histogram Pattern.

### I. Introduction

Color matching is a method in image processing of color adjustment of two images using the image features. The matching contains the two steps which are given below:-

A. **Feature Extraction:** The first step in the process is extracting image features to a distinguishable extent.

B. **Matching:** The second step involves matching these features to yield a result that is visually similar.[1]

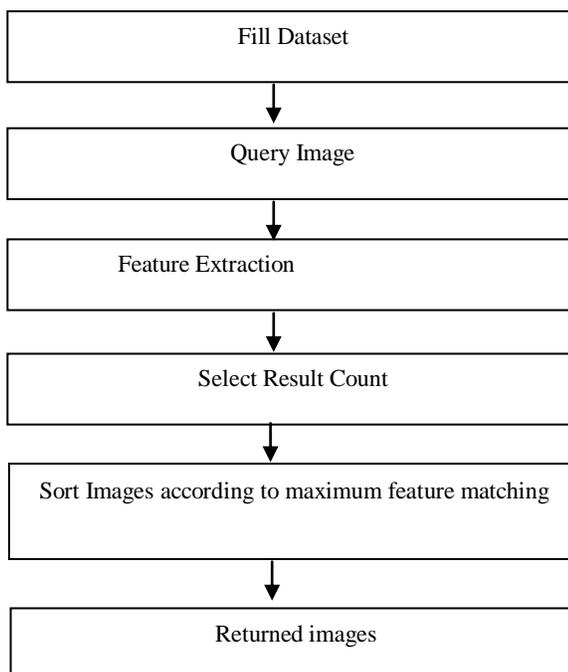
Now, in this paper we use these two steps for matching of color images. In section II we explain the methodology used. In section III contains conclusion and future scope.

### II. Dataset

In our paper, we will consider different formats of images. We take 700 images of jpeg format, 150 .png images, 150 bmp images for matching.

### III. Methodology Used And Implementation

For implementation we used the MATLAB platform.



#### A.FILL DATASET

Firstly we upload the dataset which we have created. It consists of different format of images like .jpg, .png, .bmp etc.

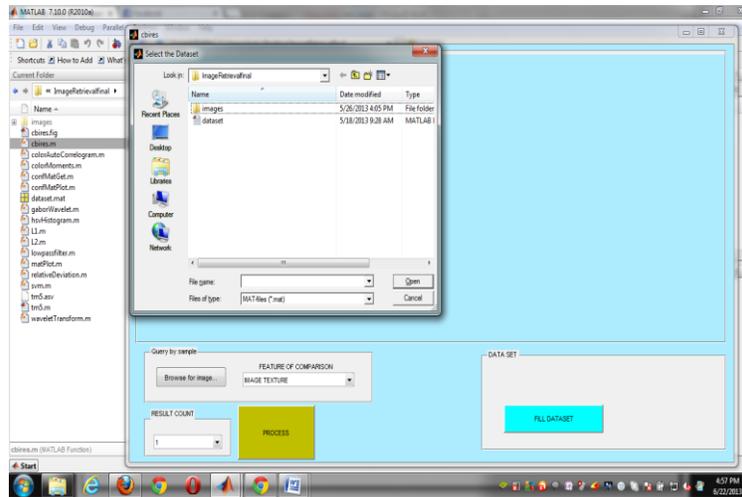


Fig. Upload Dataset

### B. QUERY IMAGE

Then we load the query image in MATLAB from the dataset which we have uploaded in the current directory.

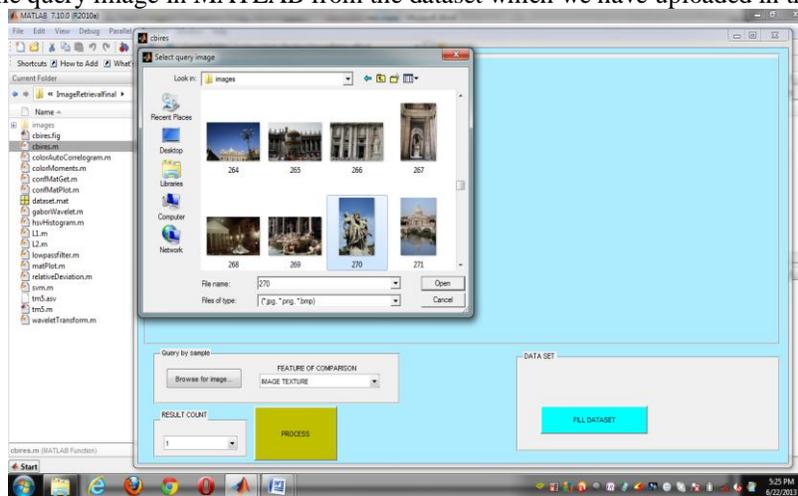


Fig. Query Image

### C. FEATURE EXTRACTION

For matching of color images we require some features so that the matching becomes easier. So, we extract some features like Image texture, Histogram Pattern[5], Size Matrix, Color contrast[4], Angle, Side lobes, RGB contrast, Shape, Relativity, Wavelet transformation, Normalized L2, Relative Deviation [3]. Matching is done for each feature.

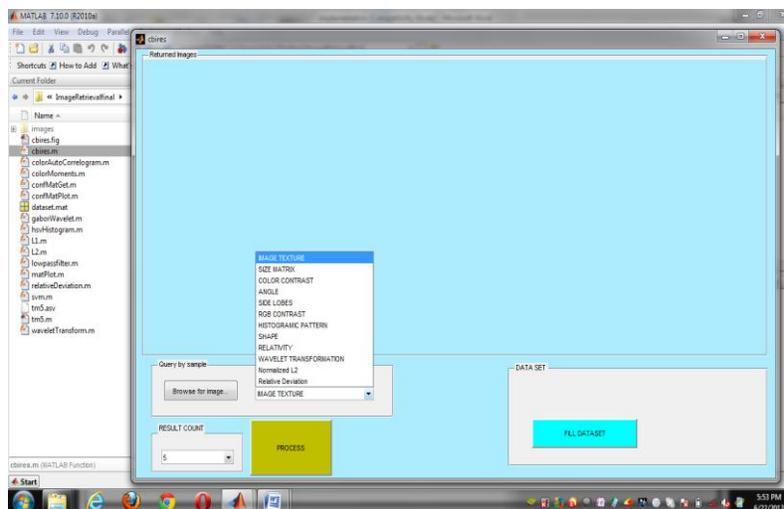


Fig. Feature Selection

#### D. RESULT COUNT

Select the result count, it will select the maximum number of images to be returned as result.

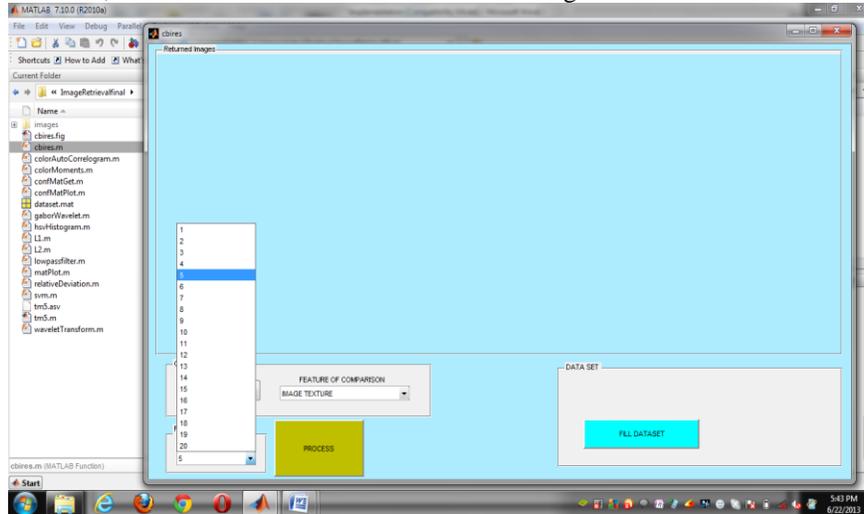


Fig. Result Count

#### E. SORTING

In this task, the goal is to sort an array of elements using the bubble sort algorithm. The elements must have a total order and the index of the array can be of any discrete type. For languages where this is not possible, sort an array of integers. The bubble sort is generally considered to be the simplest sorting algorithm. Because of its simplicity and ease of visualization, it is often taught in introductory computer science courses. Because of its abysmal  $O(n^2)$  performance, it is not used often for large (or even medium-sized) datasets.

The bubble sort works by passing sequentially over a list, comparing each value to the one immediately after it. If the first value is greater than the second, their positions are switched. Over a number of passes, at most equal to the number of elements in the list, all of the values drift into their correct positions (large values "bubble" rapidly toward the end, pushing others down around them). Because each pass finds the maximum item and puts it at the end, the portion of the list to be sorted can be reduced at each pass. A Boolean variable is used to track whether any changes have been made in the current pass; when a pass completes without changing anything, the algorithm exits [2].

#### F. RETURNED IMAGES

After sorting, Best Matched images are retrieved from the given Dataset.

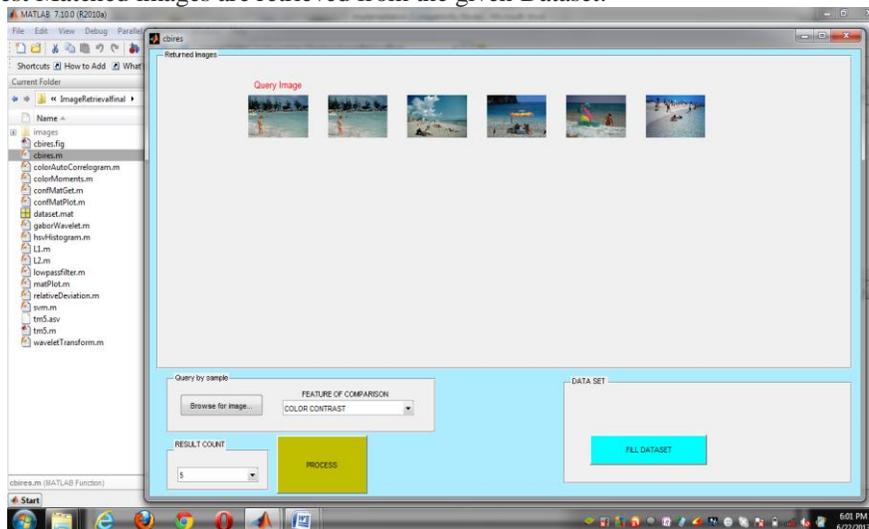


Fig. Returned images

### IV. Conclusion And Future Scope

The dramatic rise in the sizes of images databases has stirred the development of effective and efficient retrieval systems. We have researched various modes of representing and retrieving the image and found there is not any efficient method for matching different format of images like .png, .bmp. So we conclude that we can also match the other formats.

The application performs a simple color-based search in an image database for an input query image, using color histograms and different features. It then shows the best matched images and we can also check the matching on different features like shape, color contrast, texture etc. In future, the researcher can increase the dataset with different format of images like .gif, .tiff, and can also extract the different features for image matching like information gain, gabor etc.

### **References**

- [1] K Naresh Babu et. al., “Image Retrieval Color, Shape and Texture features using content based”, International Journal of Engineering Science and Technology, Volume 2, 2010.
- [2] [online] [http://rosettacode.org/wiki/Sorting\\_algorithms/Bubble\\_sort](http://rosettacode.org/wiki/Sorting_algorithms/Bubble_sort)
- [3] Peter Howarth and Stefan R`uger, “Evaluation of Texture Features for Content-Based Image Retrieval”, CIVR 2004, pp. 326–334.
- [4] [online] [https://en.wikipedia.org/wiki/Contrast\\_\(vision\)](https://en.wikipedia.org/wiki/Contrast_(vision))
- [5] Wasim Khan, Shiv Kumar, Neetesh Gupta, Nilofar Khan, “A Proposed Method for Image Retrieval using Histogram values and Texture Descriptor Analysis”, International Journal of Soft Computing and Engineering (IJSCE), Volume-I, Issue-II, May 2011.