Fast Image Retrieval Prototype using Color Descriptor

Bushra Abdul-Kareem Abdul-Azeez
Middle Technical University/Medical Institute of Technology

Abstract

In recent years, image retrieval prototypes become important and increased noticeably. Color feature is one of the most significant features to represent image. In this paper, we use a Dominant Color (DC) feature to represent images where each image represented by 8-DCs as maximum. Based on DCs values, image database is indexed using 3-D RGB partitioning color space. This is to reduce searching process where once a query image is given to the prototype; it will not search the whole database. Proposed technique will identify the partition and search the image within this partition only. According to the proposed method, extensive experiments were conducted on Corel databases. As a result, the retrieval time is reduced significantly without degradation to precision of retrieval.

Keywords: Image retrieval, RGB color space, Fast CBIR search, Image database.

I. Introduction

In the last decade, Content-Based Image Retrieval (CBIR) has become an interesting area of research. Retrieval of visual information become very important in many applications such as fashion design, crime prevention, and face recognition. Accordingly, this research field becomes one of the important and fastest growing fields in information technology. In the beginning, image retrieval used text-based image retrieval. In this method, there are many problems such as manual annotation of images and different annotators' perspectives will lead to different keywords that associated with image [1]. Due to this limitations, researchers came up with CBIR where images are retrieved based on automatically extracted low-level features (human vision related), middle-level features (objects related), or high-level features (semantic related). Among these features, the low-level features (such as color, texture and shape) are the most popular due to its simplicity [2]. One of the important low-level features is color where it plays an important role in CBIR due to its robustness to complex background and independent of image size and orientation [3-4].

Searching huge images database forces several problems which are: i) long time is needed to retrieve similar images of query; ii) low performance is resulted of image retrieval system. Additionally, there are some issues associated with CBIR performance: first one concerns with accuracy of image retrieval, where most researches in this category ignore the efficiency, they depend on sequential search [5]. Second issue is efficiency, where researchers focus on the time of image retrieval. This research will concentrate on efficiency.

Color feature is a prominent one from many existing Low Level Visual Features [6-7]. Some earlier methods are extensively utilized in CBIR systems such as dominant color descriptors (DCDs) and color Histogram [7]. Color histogram suffers from high dimensionality problem that in turn will increase its computations in similarity measure as well as make it inefficient in searching process. Therefore, dimensions reduction methods are needed.

In DCDs, maximum eight colors are extracted from the image. In this method, vector quantization (VQ) method is used. In VQ, centroid of cluster is extracted using one of clustering algorithms such as k-means, this will lead to color estimation problem. This estimation problem will reduce CBIR accuracy. This is because, images will considered as matched images when their cluster centroids are approximately equal where centroids are extracted by averaging all colors in the cluster. As noticed, instead of comparing actual color values, approximated centroids values will be compared. Accordingly, color indexing method using RGB color space is proposed in this research. Color space of RGB coloring system will separate into small parts using octal tree color quantization method [8].

An organization of this paper can be arranged as follows. Section II explains the general content based image retrieval indexing methods. As well as, it explains color-based methods. RGB-based indexing method is detailed in Section III where it focuses on the proposed technique that speeds up an image retrieval process. Section IV explains results of proposed method compared with state of art methods. Finally, the conclusion in Section V is outlined.

II. Background and Previous Works

In huge image datasets, indexing plays an important issue to decrease the search's time and space of the image retrieval (IR) operation. Most color-based methods achieve traditional sequential search during retrieval; this will enforce wasting in time of IR systems. Histogram is one of these color-based image descriptors. Even with its great characteristics such as simplicity and ease of implementation, histogram suffers from "curse of dimensionality". Therefore, Dominant Colors are utilized to elude from this problem. Consequently, Dominant Color Descriptor will be used to achieve image indexing in this research.

There are two key methods for image indexing:

1) Multidimensional indexing, and
2) Vector quantization techniques.

First indexing method partitioned into two groups, Space and Data Partitioning approaches where they partition the space or data into small parts. Example methods for these methods are kd-tree (Space Partitioning) and R-tree (data Partitioning) [9]. One of the important output of Space Partitioning approach is whole disjoints partitions of the whole space where it results partitions without overlapping between them.

In VQ method, there are several techniques, for example hierarchical K-means clustering. Output of these techniques is grouping the data into clusters where each cluster has single value called Cluster’s Centroid (CC). CC can be found by averaging of all cluster participants. In image retrieval, the query is compared to CC instead of actual values of cluster members. This will lead to inaccurate results due to many of cluster’s members have far distance from the query whereas CC is near to this query.

For color indexing, there are several methods have been proposed. In Deng et al. [10], histogram indexing is used where it suffers from high dimensional problem. In [3], color clustering and R-Tree method are used for indexing. In [11], they used LUV color clustering (by mean shift algorithm) and R*-Tree method for indexing. These previous methods based on color clustering that suffers from color estimation problem of clustering [10]. Recently, researchers in [9] contributes to overcome this problem where it introduce two threshold variables to reduce error in similarity to CC. Accordingly, this paper discuss this problem and propose solution to avoid it. All previous related works can be summarized in Table1.

Table 1: Related Works Overview Table

Author(Year)	Technique/Method	Description	Drawbacks
Babu et al. [2]	color clustering and spatial indexing (R-Tree)	Combines color clustering and spatial indexing method for indexing colors of flags and trademarks databases	compute clusters' centroids instead of original values lead to inaccurate results
Deng et al. [9]	Histogram Indexing	Use histogram entries to index colors and images in database	High dimensional indexing (1024-D of color histogram bins)
Sudhamani and Venugopal [10]	Mean-shift and R*-Tree spatial indexing	Use mean shift algorithm for color clustering, R*-Tree for spatial indexing and LUV color space instead of RGB	compute clusters' centroids instead of original values lead to inaccurate results
Yildizer et al. [8]	An-improved K-means	This research introduce two threshold values that can be considered as search space parameters	High overhead for extracting the two search parameters

III. Proposed Color Indexing Method

The purpose of proposed indexing technique is speeding up retrieval process in CBIR without significant degradation in retrieval quality. But how can do that? The answer simply is looking for images of matched colors to the query image. Thus, only images of colors that have distance less than specific threshold from query's DCs will be searched.

To construct DB index structure, color similarity threshold must be determined by Euclidian distance where it will be set to 25 of according to three channels of RGB.

- **Indexing Structure**

In proposed method, DCs are extracted using dynamic quantization method. In this case, image is quantized to five-bits (from 3^{rd} to 7^{th} bits) of each color band. As the difference between two color bands is 25, the range of the two bits (3^{rd} and 4^{th}) satisfies this difference. To decide two colors are similar or not, the last three bits (5^{th}, 6^{th} and 7^{th}) are far. Thus, these bits are exploited to be the first similarity level in the proposed indexing method; they will use to discriminate similar colors from not similar. The 3^{rd} and 4^{th} bits are used individually to be the subsequent levels of similarity (2^{nd} and 3^{rd} levels). Thus, first level of index structure will have 512 cells. The 2^{nd} and 3^{rd} levels of indexing structure will have eight cells. The 4^{th} level contains 4 entries that represent color ratio of certain DC in its image; each entry has 25% size of image. This is the key difference over research in [12]. Figure 1 represents our indexing scheme.

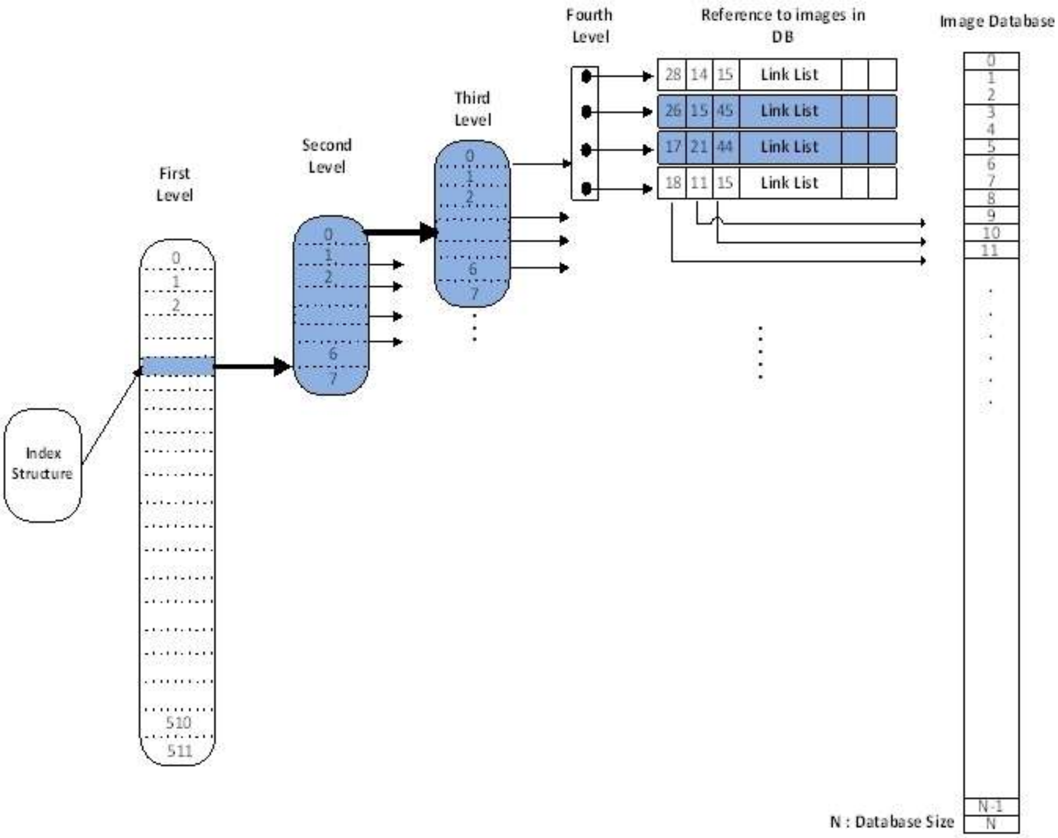


Figure 1: An Indexing scheme

In [10], reducing search space is achieved through query submission, this will lead to that not similar images will not be considered in searching process. It is not practical in real applications like online image retrieval. This will consume time especially in huge database. Therefore, this research perform it offline to speed up query processing.

- **Image Searching**

In searching process, a query is submitted to CBIR prototype that will search on the images that are similar to it in DB. Searching Algorithm comprises some steps as follows:

Color Indexing-based Searching Algorithm

Input: Image Database; Query Image

Output: Images that are Similar to Query Image

```
Step1 : [For each DC in the query image]
        DCs= Find_DominantColors ( QueryImage)
        For i= 0 to DCs-1
                [Step 2 inside this loop]
Step 2: [find database images that are similar in both colors values]
        For j= 0 to ImgDB.lenght-1
        If DC(i) = ImgDB_DC(j)  then
                [Step 3 inside this loop]
Step 3: [Reaching to the nodes in the third level of index structure]
        If Percentage(DC(i)) = Percentage(DC(ImgDB_DC(j))) then
                [Step 4 inside this If]
Step 4: [Merging images references that resulted from each DC of the query image
to produce search space (RSS) of the query]
        Add (ImgDB(j), MatchedList)
Step 5: [Calculate dissimilarity distance between query and all images in the RSS]
        SM_List ← SimilarityMeasure(QueryImg, MatchedList)
Step 6: [Rank them according their Similarity measure (SM)]
        Sort (SM_List, MatchedList)
Step 7: Return Top N Similar Images
        RetList ← Empty
        For i= 0 to N-1
        RetList.add (MatchedList(i))
        Next i
            Return (RetList)
Step 8: End.
```

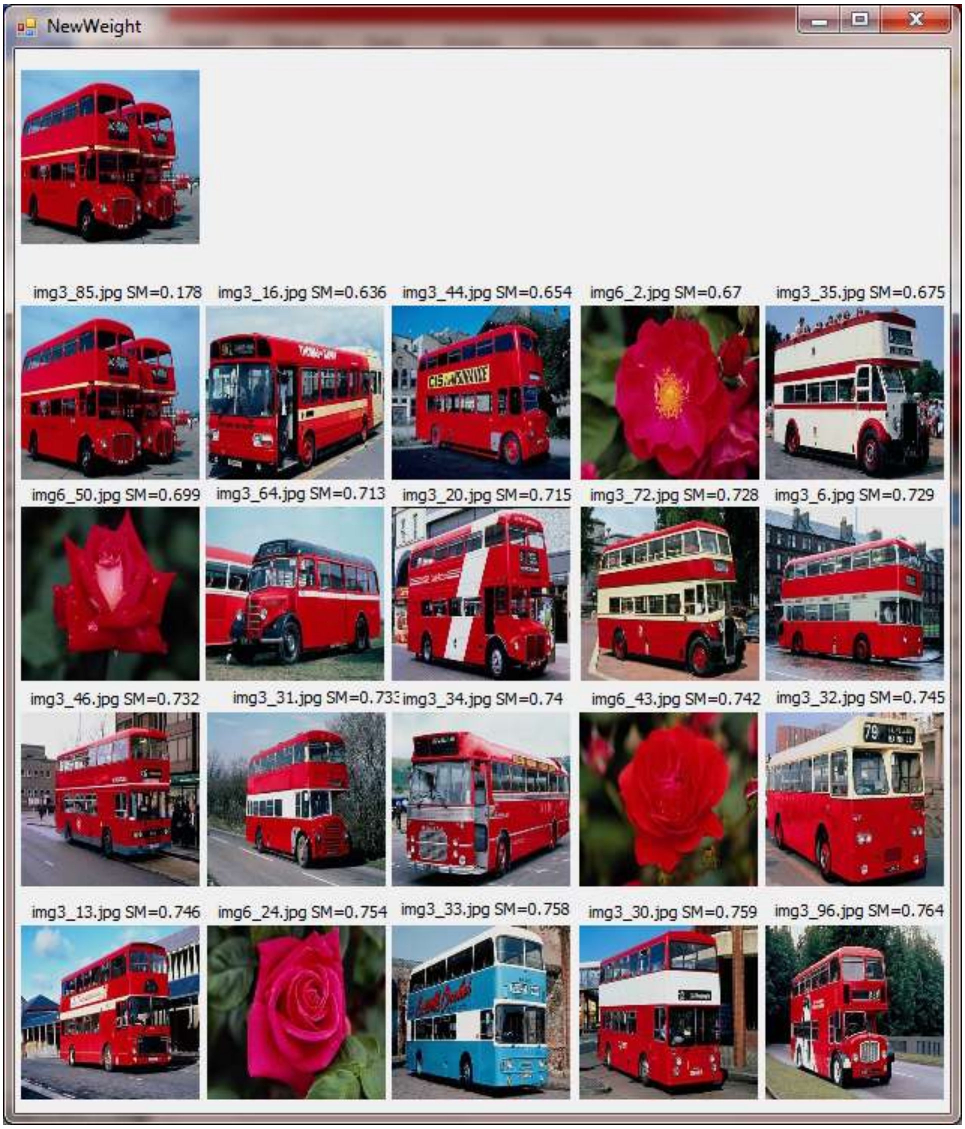
In aforementioned steps, most not match images are filtered out according to large distance value among colors. According to max distance value among colors, three color allowance values (CAVs) are used. These CAVs are 0, 8 and 24 as will explain in section IV.

IV. Experimental Results

Experimental setup need to be specified first before displaying the results as follows:

- Indexing will be achieved with depends on (8, 5, 3 or 1) colors. This is because to measure indexing performance on each of the number of colors. Definitely, the largest percentage DCs will be selected.
- To speed up image database's searching, we need to alleviate depending on sequential search. This can be achieved by reducing number of compared images with query. The reduced time of proposed method (***Search Space Ratio (SSR)***) can be computed by Reduced Search Space (RSS) and Whole Search Space (WSS) by using following formula:

  SSR= RSS/WSS
- Two quantitative metrics are used to compute the accuracy of proposed indexed dominant color descriptor. These metrics are Precision(10) (denoted as Prc(10)) and Average Retrieval Rate (denoted as ARR).


NewWeight
img3_85.jpg SM=0.178
img3_16.jpg SM=0.636
img3_44.jpg SM=0.654
img6_2.jpg SM=0.67
img3_35.jpg SM=0.675
img6_50.jpg SM=0.699
img3_64.jpg SM=0.713
img3_20.jpg SM=0.715
img3_72.jpg SM=0.728
img3_6.jpg SM=0.729
img3_46.jpg SM=0.732
img3_31.jpg SM=0.73
img3_34.jpg SM=0.74
img6_43.jpg SM=0.742
img3_32.jpg SM=0.745
img3_13.jpg SM=0.746
img6_24.jpg SM=0.754
img3_33.jpg SM=0.758
img3_30.jpg SM=0.759
img3_96.jpg SM=0.764

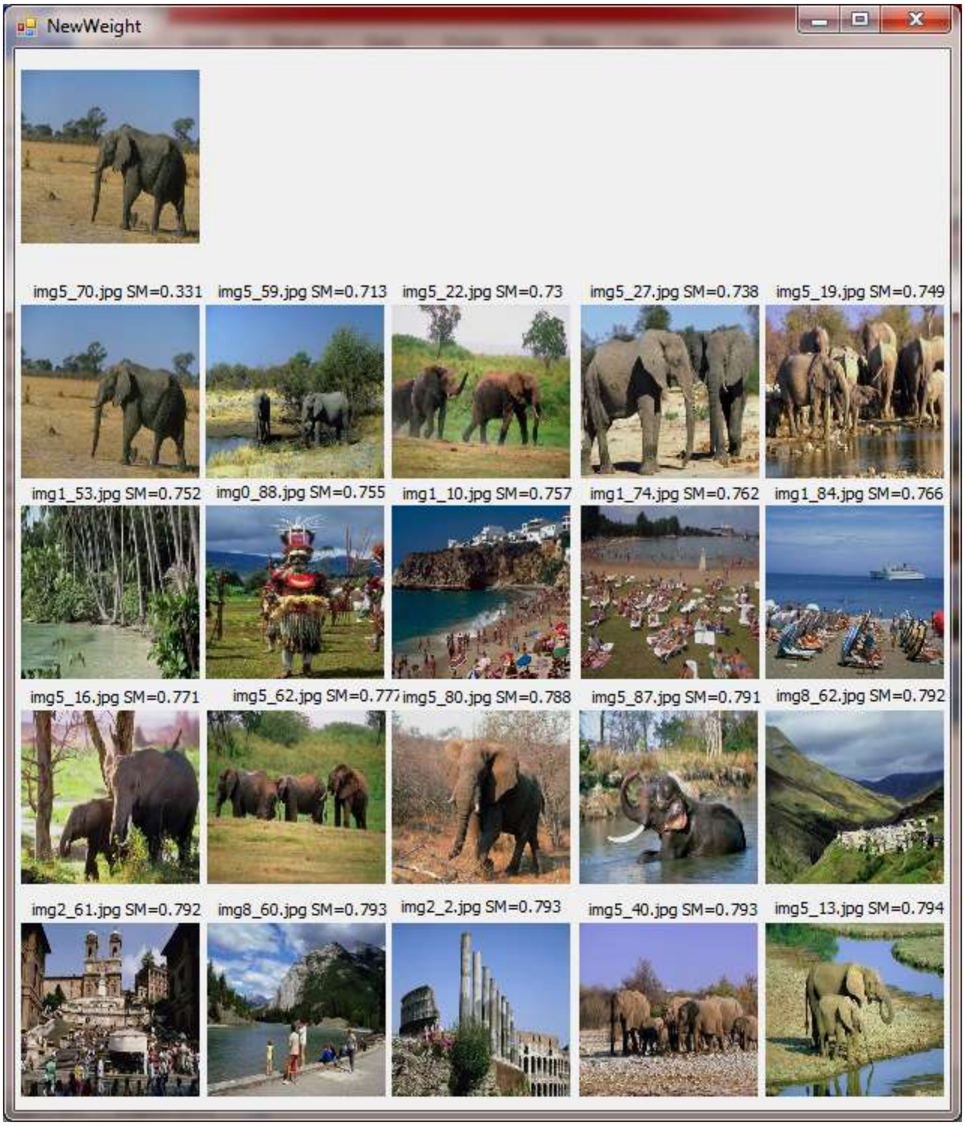


Figure 2: Visual Results of DCD with Proposed Indexing Method in Corel-10K Dataset

- To evaluate the proposed indexing scheme, Corel dataset (10,800 images) is used. To test color descriptor, MPEG-7 DCD will be used to check proposed indexing method.

<u>Table 2:</u> ***Search Space Ratio*** in DCD using sequential and proposed indexed searching in Corel Dataset

Indexed Colors	8-Colors	5-Colors	3-Colors	1-Color
DCD	**Search Space Ration**	**Search Space Ration**	**Search Space Ration**	**Search Space Ration**
Sequential Search	100%			
Proposed Method CAV =24	**57.3%**	**49.8%**	**40.5%**	24.5%
Proposed Method CAV=8	**27%**	**25%**	**22.1%**	16.3%
Proposed Method CAV=0	14%	13.6%	12.8%	10.8%

<u>Table 3:</u> ***Search Accuracy*** in DCD using sequential vs. proposed indexed searching in Corel Dataset

Indexed Colors	8-Colors	5-Colors	3-Colors	1-Color
DCD	**Prc(10) ARR**	**Prc(10) ARR**	**Prc(10) ARR**	**Prc(10) ARR**
Sequential Search	0.230/ 0.060			
Proposed Method CAV =24	**0.230/ 0.060**	**0.240/ 0.061**	**0.240/ 0.059**	0.230/ 0.057
Proposed Method CAV=8	**0.240/ 0.060**	**0.240/ 0.061**	**0.240/ 0.060**	0.230/ 0.056
Proposed Method CAV=0	**0.230/ 0.060**	**0.230/ 0.061**	0.230/ 0.058	0.200/ 0.054

From Table 2 and 3, we note that the proposed octal tree indexing method Reduce the search space approximately to 25% from the whole search space (100%) that used in sequential search method with increasing (or at least preserving) the accuracy value of the DCD. This certainly will speed up the searching process.

V. Conclusion

In this research, fast image retrieval is proposed. This is achieved by indexing images depending on their DCs. In this research, two indexing problems are overcome, by proposing indexing method that depends on dominant colors. The proposed indexing method outperformed over the state of art methods by using DC for indexing using octal tree quantization method. The characteristic of the proposed indexing scheme is it can be utilized for diverse color descriptors. Additionally, the suitable design of proposed index structure offers high speed retrieval.

References

[1] Yang, L.F., Huang, X.L., Lv, R., and Lv, H.: 'An Effective Similarity Measurement Algorithm for Dominant Color Feature Matching in Image Retrieval', Applied Mechanics and Materials, 2012, 182, pp. 1169-1173

[2] Zhang, J., and Zou, W.: 'Content-Based Image Retrieval using color and edge direction features', in Editor (Ed.)^(Eds.): 'Book Content-Based Image Retrieval using color and edge direction features' (IEEE, 2010, edn.), pp. 459-462

[3] Babu, G.P., Mehtre, B.M., and Kankanhalli, M.S.: 'Color indexing for efficient image retrieval', Multimedia Tools and Applications, 1995, 1 (November), pp. 327–348

[4] Pass, G., and Zabih, R.: 'Histogram refinement for content-based image retrieval', in Editor (Ed.)^(Eds.): 'Book Histogram refinement for content-based image retrieval' (IEEE, 1996, edn.), pp. 96-102

[5] Amsaleg, L., Gros, P., and Berrani, S.-A.: 'Robust object recognition in images and the related database problems', Multimedia Tools and applications, 2004, 23, (3), pp. 221-235; Amsaleg, L., and Gros, P.: 'Content-based retrieval using local descriptors: Problems and issues from a database perspective', Pattern Analysis & Applications, 2001, 4, (2-3), pp. 108-124

[6] Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., and Yamada, A.: 'Color and texture descriptors', Circuits and Systems for Video Technology, IEEE Transactions on, 2001, 11, (6), pp. 703-715

[7] Van De Sande, K.E., Gevers, T., and Snoek, C.G.: 'Evaluating color descriptors for object and scene recognition', Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 32, (9), pp. 1582-1596

[8] Gervautz, M., and Purgathofer, W.: 'A simple method for color quantization: Octree quantization': 'New trends in computer graphics' (Springer, 1988), pp. 219-231

[9] Yildizer, E., Balci, A.M., Jarada, T.N., and Alhajj, R.: 'Integrating wavelets with clustering and indexing for effective content-based image retrieval', Knowledge-Based Systems, 2012, 31, pp. 55-66

[10] Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., and Shin, H.: ' An efficient color representation for image retrieval', IEEE Trans. Image Process, 2001, 10 (1), pp. 140–147

[11] Sudhamani, M., and Venugopal, C.: 'Grouping and indexing color features for efficient image retrieval', International. Journal of Applied Mathematics and Computer Sciences. v4 i3, 2007, pp. 150-155

[12] Talib, A., Mahmuddin, M., Husni, H., and George, L.E.: 'Dominant Color-Based Indexing Method for Fast Content-Based Image Retrieval': 'Recent Advances in Information and Communication Technology' (Springer, 2014), pp. 135-144