Image Retrieval using One-Dimensional Color Histogram Created with Entropy

Mahmut KILIÇASLAN[1], Ufuk TANYERİ[1], Recep DEMİRCİ[2]
[1]*Ankara University, Nallihan Vocational School, Computer Technologies Department, Turkey*
[2]*Gazi University, Faculty of Technology, Computer Engineering Department, Turkey*
m.kilicaslan@ankara.edu.tr

***Abstract*—Image histograms are frequently used as a feature vector in content-based image retrieval (CBIR). The related methodology involves processing of a single channel histogram on gray level images while histograms of three channels must be processed in color images. Subsequently, there are two ways to process histograms of color images. In the first approach, the length of feature vector is extended by adding histogram data of each channel to create new feature vector. However, this kind of solution increases computational time and complexity. Second solution is to combine the histogram data obtained from each channel to establish a feature vector. In this study, a novel image retrieval approach, which uses a cluster-based one-dimensional histogram (ODH) for color images has been developed. Initially, multiple thresholds (MT) for each channel were calculated by means of Kapur entropy method. Then, the RGB color space was subdivided into sub-cubes or prisms. The numbers of pixels in each cluster and cluster index or class label have been used to construct a cluster-based one-dimensional histogram. Finally, image retrieval process has been implemented by using the one-dimensional color histogram (ODH) of images in database and query.**

***Index Terms*—entropy, feature extraction, histograms, image retrieval, vector quantization.**

I. INTRODUCTION

The number of digital images and their qualities with the development of technology in recent years, and consequently their usage areas has increased. The image retrieval has a critical importance because of the fact that today, the users reach to more images via the internet than in the past. However, it is not easy to retrieve the desired image among from numerous images in databases [1, 2]. In other words, retrieval strategies developed so far still need to be improved for especially color image queries.

Initial studies about the relevant issue in the 1970s created a solution in the form of indexing images and identifying them with keywords. The determination of keywords was made by means of dictionaries and query words, which were typed by the users. Nevertheless, keywords could always not be selected correctly since human perception varied from person to person. Additionally, a large amount of operator effort was required to carry out the process regularly.

Due to the difficulties of text-based image retrieval systems that was introduced in the 1970s, CBIR approaches were developed in the 1980s. A typical CBIR system consists of three basic stages: feature extraction, image database and similarity measure as shown in Fig. 1. The image database is collection of conceptually relevant images such as art collections, criminal, medical or military applications. The CBIR systems are fundamentally based on extracting features from images. Also the overall performance of the CBIR system must be evaluated by using additional metrics. The high precision and short retrieval period are key requirements of the CBIR system [3-12].

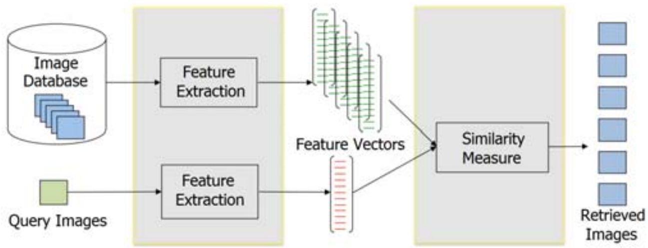

Figure 1. A typical CBIR system and its stages [6]

In CBIR systems, the feature extraction stage is the most important step. The representative abilities of the attained features for any image are critical for success of CBIR system. If the query image and any image in database are not correctly defined in terms of feature vector, the accuracy of the system will be low as a result. Moreover, length of feature vector and computational cost of feature extraction stage are also important for retrieval time of the systems. Therefore, researchers focus on increasing performance of content based image retrieval systems.

Atto et al. proposed a CBIR system in their study, which utilizes the pattern properties, which were obtained from sub-bands of wavelet packet in a heterogeneous data set [13]. On the other hand, Zhao et al. extracted the geometric pattern by using the maximum entropy principle for a similar data set. Nevertheless, there is a kind of disadvantage that the patterns within salient regions are obtained by iterative processes as it increases computation time [14]. Ashraf et al. considered a geometric pattern of image as a feature. In their approach, the features extracted by using Bandelet transformation, artificial neural network, and Gabor filter were combined with RGB-YCbCr conversion. Furthermore, Canny edge detector and discrete wavelet transform were used [15]. Consequently, pre-processes in their applications increase computational complexity.

Lu et al. suggested a CBIR method based on entropy and relevance feedback with HSV color space transformation. The technique developed based on converting multiple features into a single feature by iteratively weighting with different similarity measurements [16]. Apart from patterns and shapes in feature extraction stage, color information in

Digital Object Identifier 10.4316/AECE.2020.02010

image is a frequently used feature since color has a very rich representation ability and is a strong descriptor. Recently, several approaches in which the color information in image was used as a feature, have been proposed. Color structure [17], dominant color [18], color layout [19], color coherence vector [20], color moments [21], and vector quantization [22] are typical descriptors which use color as a feature. Image histogram is a simple, effective and frequently preferred global representation tool that identifies images with color levels and their number [23-26]. In other words, content of images could be defined by means of histogram. In gray level images, the histogram is expressed by a one-dimensional vector whereas three different histograms must be created in color images. Therefore, combining histogram information from each color channel is an issue to be solved. If color histogram is used for retrieval, a three-dimensional matrix must be processed. However, computational cost will be quite high since there are 2^{24} different colors in RGB color space.

Thus, color quantization approaches are needed for image retrieval. The color quantization process can be defined as the aggregation of similar colors under the same group [27] and Linde-Buzo-Gray (LBG) algorithm is a frequently used approach as vector quantization [28]. Nevertheless, the LBG algorithm is also used in several process such as clustering [29], fast search [30], image compression [31], codebook generation [32], speech processing [33] and CBIR system [34]. Wherever it is used, the LBG method requires an initialization of cluster center randomly. The algorithm iteratively searches correct position of cluster centers until predefined conditions are satisfied.

In this study, a new color reduction or vector quantization (VQ) algorithm has been proposed. It is based on multi-level thresholding of color images. Multiple thresholds for each channel were calculated by means of Kapur entropy algorithm. Then, the RGB color space was partitioned into sub-cubes or prisms. Pixels located in each prism in RGB color space are assigned into the same cluster. Subsequently, averages of pixels of related cluster are substituted into their spatial positions. As the number of prisms created depends on number of thresholds used, the number of colors in the output image will be the same as cluster number. As each sub-cube in RGB color space has index, every reduced color in output image will have the same index. Hence, a cluster-based one-dimensional histogram (ODH) for color images has been defined by using cluster index and relevant bins. Finally, image retrieval process has been implemented by using the one-dimensional color histograms of images in database and query.

II. Color Quantization

Color quantization, which is generally used in clustering and compression, is a typical kind of vector quantization. The vector quantization is reduction of number of vectors so that a data set could be represented by vectors with limited numbers. A pixel in color image is vector in color space. Therefore, the number of vectors in a color image is the same as number of pixels. When the color quantization algorithm is applied for color image, the size of image is not changed, only kind of colors in image is reduced. In other words, pixels in image are classified into meaningful subsets. Subsequently, a look-up table or codebook is created. The color reduction unavoidably results in information losses in original data. Therefore, the quality of the reduced image with the related codebook is measured by the representation capability of the original image [35, 36]. The performance of color reduction depends on algorithm used and the number of cluster employed. Furthermore, it is expected that any color reduction algorithm produces the same results. Although several color reduction algorithms have been proposed so far, the LBG algorithm is frequently preferred approach for color quantization [37, 38].

A. LBG algorithm

LBG algorithm is an iterative vector quantization algorithm, which is frequently used for color quantization. It reduces any d-dimensional input vector set, $X = \{x_i \in R^d \,|\, i = 1, 2, \ldots, n\}$ to an output vector set with k elements, $C = \{c_j \in R^d \,|\, j = 1, 2, \ldots, k\}$ by using similarity criteria. The output vector set is also called the codebook. As it is an iterative process which is carried out until pre-defined condition is reached. Nevertheless, number of cluster, k, number of iterations, t and threshold value, ε must be selected by user initially. D_t which is amount of distortion in iterations, t must also be evaluated. The fundamental problem with LBG is that the initial values of codebook are randomly selected. Consequently, the output of the algorithm may be different in each run. Steps of the LBG algorithm could be define as follow:

Step 1: Get X input vector set,

Step 2: Generate C codebook randomly,

Step 3: Let $D_0 = 0$ and $t = 0$,

Step 4: Put x_i in partitioned set F_j if $\|x_i - c_j\| \le \|x_i - c_q\|$ for $j \ne q$, can be separated into k sets where F_j denotes $F_j = \{x_1^j, x_2^j, \ldots, x_{V_j}^j\}$, and V_j is total number of data vector set in S_j.

Step 5: Update $c_j \,|\, j = 1, 2, \ldots, k$, cluster centers according to $c_j = \frac{1}{|F_j|} \sum_{x_i \in F_j} x_i^j$,

Step 6: Calculate distortion, $D_t = \sum_{j=1}^{k} \sum_{x_i \in F_j} \|x_i - c_j\|$,

Step 7: Repeat steps 4-6, if $\frac{D_{t-1} - D_t}{D_t} > \varepsilon$,

Step 8: If not, give the codebook C as output in the last state.

B. Entropy based multi-level thresholding

Image thresholding is one of image clustering process, which is grouping similar pixels of image under same label. The thresholding is based on image histogram. When the number of threshold increases, the homogeneity of cluster created will be high. Entropy based approach is a widely used thresholding technique developed by Kapur [39, 40]. Gray scale image has only one channel and therefore single histogram is calculated. On the other hand, an intensity distribution curve for every channel must be calculated as color image has three channel. Intensities in image are discretized and each level is indexed by i $(0,1,\ldots,L-1)$ where L is maximum level. Consequently, the probability of i^{th} level is calculated as

$$p_i = n_i / (M \times N) \tag{1}$$

where n_i is number of pixels in i^{th} level and $M \times N$ is the size of the image. Threshold values for image clustering are estimated by means of probability distribution given in (1). When multi-level thresholds as $t_1, t_2, \ldots, t_m$ for gray scale image are selected, the number of cluster formed will be $m+1$ and intensity ranges of clusters are assigned as follows:

$$\begin{aligned} c_0 &= \{0,\ldots,t_1-1\} \\ c_1 &= \{t_1,\ldots,t_2-1\} \\ &\ldots, \\ c_m &= \{t_m,\ldots,L-1\} \end{aligned} \tag{2}$$

The fundamental problem with thresholding is determination of relevant thresholds. In entropy based multi-level thresholding, an objective function depending on with threshold variables is chosen as

$$J(t_1,t_2,\ldots,t_m) = H_1 + H_2 + \ldots + H_m \tag{3}$$

where $H_1,\ldots,H_m$ are partial entropies of each class and they are defined as

$$\begin{aligned} H_1 &= -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t_1-1} p_i \\ H_2 &= -\sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \omega_1 = \sum_{i=t_1}^{t_2-1} p_i \\ &\vdots \\ H_m &= -\sum_{i=t_m}^{L-1} \frac{p_i}{\omega_m} \ln \frac{p_i}{\omega_m}, \omega_m = \sum_{i=t_m}^{L-1} p_i \end{aligned} \tag{4}$$

where $\omega_0,\ldots,\omega_m$ are partial probabilities of clusters. The values of variables, which make (3) maximum are considered to be accurate thresholding for gray scale image clustering.

C. Multilevel thresholding and one-dimensional color histogram

Histogram is a kind of feature vector of gray scale image. In gray scale image retrieval, it is basically satisfactory approach to be used. On the other hand, as color image has three channels, three different histograms must be either processed or combined in color image retrieval. Second solution for color image retrieval is to used color histogram of which computation cost is quite high as three dimensional array must be processed. Furthermore, if dimensions of feature vectors increases, memory requirements for saving feature vectors also rise apart from computation time.

In this study, a novel approach was developed for color image retrieval where one-dimensional histogram (ODH) of color image was used. In fact, the ODH based image clustering has been recently proposed by authors [41]. In the ODH procedure, initially, RGB color is partitioned by using threshold values obtained for each channel. In other words, three dimensional color space is sliced so that sub-cubes or prisms could be created. Then, pixels located in each sub-cube in color space are assigned into the same cluster. Furthermore, a new color reduction algorithm was devised by means of average of pixel in the same cluster. Homogeneity of clusters are related to volume of sub-cubes created in color space. Additionally, similarity of original images and the images obtained as result of color reduction depends on sizes of sub-cubes. Therefore, the volume of sub-cubes created in color space was decreased by means of multi-level thresholding proposed by Farshi and et al. [42]. In their approach, multiple thresholds have been obtained by using optimization algorithms. The number of sub-cubes in the color space is $k=(m+1)^3$ where the number of threshold was chosen as m. Subsequently, the labels of clusters are distributed as $\{c_0, c_1, \ldots, c_{k-1}\}$. Fig. 2(a) shows the subdivision of the color space with single threshold (ST) where there are eight sub prisms whereas the results of partition with two thresholds is shown in Fig. 2(b). The number of clusters becomes 27 with two thresholds.

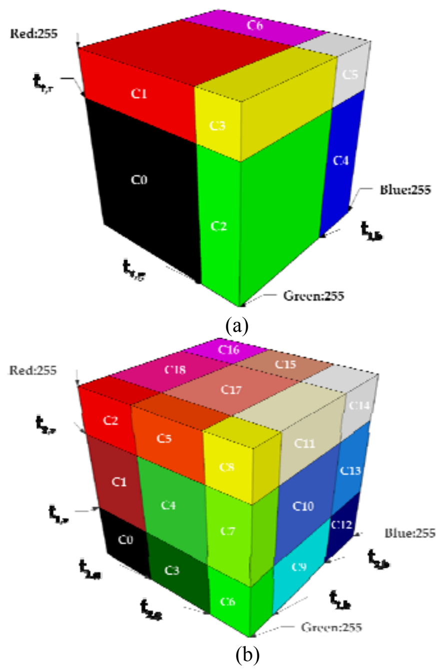

Figure 2. Partition RGB color space (a) $m=1$, $k=8$ (b) $m=2$, $k=27$.

In order to slice the color space with a single threshold, the required rules and class labels are given in Table I where t_r, t_g and t_b are threshold values calculated for red, green and blue channels respectively. Moreover, r_i is number of pixels assigned into i^{th} class. Since the total number of pixels in an image is equal to sum of the number of pixels in all classes, following relation,

$$M \times N = \sum_{i=0}^{7} r_i \tag{5}$$

could be written. Accordingly, for probability of any pixel being in i^{th} class, it could be written as follows:

$$p_i = \frac{r_i}{M \times N} \tag{6}$$

Consequently, the probability distribution function, p_i is considered as a single dimensional histogram of color image. Thus, it is possible to represent color images without using three different histograms. As the averages of each class is replaced in corresponding pixel position, vector quantization or color reduction process is simultaneously realized. As a result, a codebook is also created. Since the cluster or prism index, i is systematically assigned in color space, their positions in color space are always fixed. Fig. 3(a) shows Peppers image and Fig. 3(b) shows its distribution in color space. In addition, its single dimensional histogram obtained with single threshold is shown in Fig. 3(c). The ODH shown in Fig. 3(c) is assessed as a feature vector for image retrieval. The dimension of feature vector depends on cluster number, k which is function of m, threshold number. Thresholds were estimated by means of (3) defined for the multi-level thresholding based on entropy. When two and three thresholds were used, the number of cubes or class will be 27 and 64, respectively.

TABLE I. PARTITION OF COLOR SPACE BY SINGLE THRESHOLD

Class label	Partition rules	Number of pixels	Binary Code
c_0	$if\ (R<=t_r\ \&\ G<=t_g\ \&\ B<=t_b)$	r_0	000
c_1	$if\ (R<=t_r\ \&\ G<=t_g\ \&\ B>=t_b)$	r_1	001
c_2	$if\ (R<=t_r\ \&\ G>=t_g\ \&\ B<=t_b)$	r_2	010
c_3	$if\ (R<=t_r\ \&\ G>=t_g\ \&\ B>=t_b)$	r_3	011
c_4	$if\ (R>=t_r\ \&\ G<=t_g\ \&\ B<=t_b)$	r_4	100
c_5	$if\ (R>=t_r\ \&\ G<=t_g\ \&\ B>=t_b)$	r_5	101
c_6	$if\ (R>=t_r\ \&\ G>=t_g\ \&\ B<=t_b)$	r_6	110
c_7	$if\ (R>=t_r\ \&\ G>=t_g\ \&\ B>=t_b)$	r_7	111

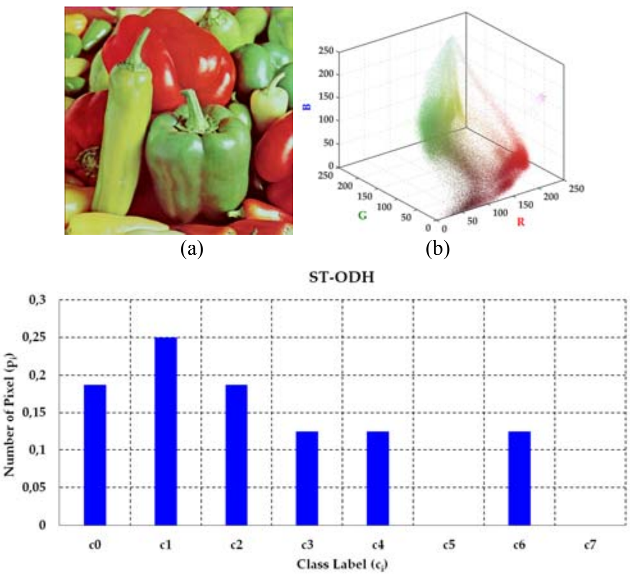

Figure 3. Peppers (a) original (b) color distribution (c) ODH with single threshold

In proposed study, RGB color space is divided into 27 and 64 subclasses by means of Kapur objective function defined in (7). For examples, Fig. 4(a) shows Peppers image quantized by using two threshold or 27 clusters. Moreover, the color distribution of Peppers reduced with 27 clusters or sub-cubes in color space is shown in Fig. 4(b). The sizes of balls in color space is proportional with numbers of pixels assigned related cluster. Fig. 5(a) shows conventional channel histograms whereas the Fig. 5(b) shows single dimensional color histogram with 27 clusters. If the ODH in Fig. 5(b) is used for image retrieval, the features vector size will be 27. In fact, the ODH in Fig. 5(b) represent quantized Peppers images. The question here is the representation capability of quantized image. The information losses are indispensable in color reduction. However, the amount of information losses could be tested by image similarity measure or peak signal to noise ratio (PSNR). The PSNR values are calculated for each channel individually. In our experiments, the average of three channels was used for evaluation. The color reduction process was repeated for 1, 2 and 3 thresholds. Consequently, PSNR values shown in Table II have been obtained. As could be seen, the similarity of reduced image increases with number of thresholds. In fact, the homogeneity of clusters or pixel locating in sub-cubes rise with threshold. Subsequently, overall similarity of image grows. In other perspective, the representation capabilities of reduced images become high.

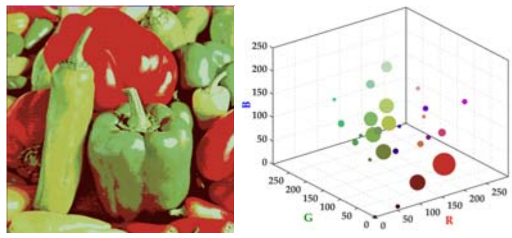

Figure 4. Pepper quantized (a) reduced, $m=2$ (b) reduced color distribution

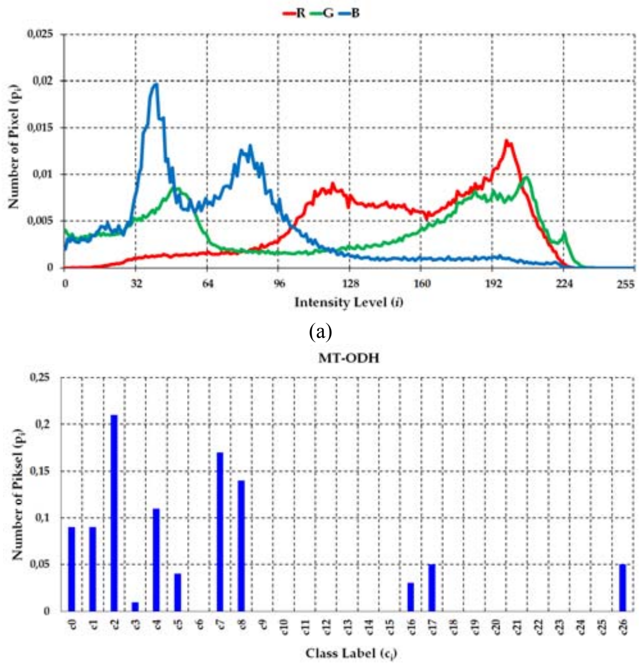

Figure 5. Pepper histograms (a) three channels (b) ODH with 27 clusters

TABLE II. PSNR VALUES OF REDUCED PEPPER IMAGE

Number of Thresholds	R	G	B	Average
1	18,408	20,556	22,660	20,541
2	24,085	24,328	26,5809	24,998
3	26,735	25,880	28,086	26,900

III. Image Retrieval with One Dimensional Histogram

Real image databases are consisting of various gray scale and color images. In CBIR systems, it is expected that true image is fetched whatever an image is queried. On the other hand, researches on image retrieval focus on either gray scale or color image datasets. Retrieval procedure in color image is harder than gray scale image datasets since processing feature vector obtained from color image requires highly computation cost. Conventionally, one dimensional histogram or feature vector is used in gray scale images. On the other hand, three different histogram or feature vector must be processed to estimate color image similarities.

In this study, a new content-based image retrieval algorithm by using one-dimensional color histogram has been developed. Fig. 6 shows block diagram of proposed strategy where feature extraction stages are replaced with entropy based multi-level thresholding and one-dimensional color histogram. Thus, image similarity measurements could be done by using any similarity technique based on image histograms. The length of feature vector could be changed by number of threshold, m. Consequently, overall performance of CBIR devised is flexible and controlled by user. Additionally, there is no need to process three different histogram or three dimensional color histograms. Therefore, it is computational cost effective as well.

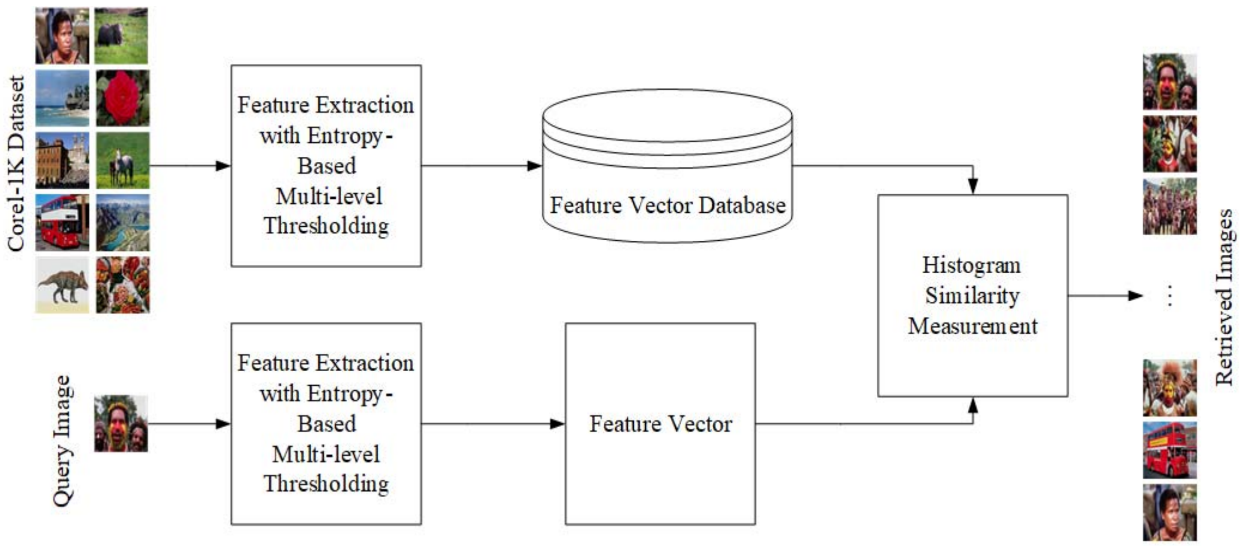

Figure 6. Proposed CBIR system based ODH.

In image similarity's estimation stage of the developed CBIR system, any similarity measurement method could be used. Since histogram intersections and cosine similarity techniques are widely used measurement tools and are parallel to human perception [5], the related two approaches were used in experiments. Histogram intersection is an approach capable of performing partial matches. Hence, similarity of the queried image and any image in database is computed as follows

$$S_I\left(H_\alpha, H_\beta\right) = \sum_{i=0}^{k} \min\left(H_{\alpha,i}, H_{\beta,i}\right) \quad (7)$$

Where H_α and H_β represent histograms of the queried image and any image in database, respectively. The related cosine similarity is calculated as follows

$$S_C\left(H_\alpha, H_\beta\right) = \frac{\sum_{i=0}^{k} H_{\alpha,i} H_{\beta,i}}{\sqrt{\sum_{i=0}^{k} H_{\alpha,i}^2} \sqrt{\sum_{i=0}^{k} H_{\beta,i}^2}} \quad (8)$$

IV. Experimental Results and Discussion

A user interface was developed in C#.NET environment for assessment of devised strategy. The Corel1K data set has been chosen to test performance of the proposed CBIR. The database consists of 10 classes with 100 images. The results of developed strategy have been compared with results of LBG algorithm. In the developed study, depending on number of thresholds: 1, 2, and 3, the number of clusters in quantized images become 8, 27 and 64, respectively. Therefore, the class numbers in LBG algorithm were selected in parallel manner for consistency.

The efficiency of the tested approaches with each similarity criterion defined in (7) and (8) have been evaluated by frequently preferred precision-recall (P-R) [43] and mean average precision (mAP) [16] given in (11). Precision is calculated as follows:

$$P = \frac{\textit{number of retrieved and relevant}}{\textit{number of retrieved}} \quad (9)$$

while the recall is calculated as

$$R = \frac{\textit{number of retrieved and relevant}}{\textit{number of relevant in collection}} \quad (10)$$

Additionally, the mAP is calculated as

$$mAP = \frac{1}{Z}\sum_{j=1}^{Z} \frac{1}{Q_j}\sum_{i=1}^{Q_j} P\left(I_i\right) \quad (11)$$

where Q_j is number of relevant images for query j, Z is the number of queries, $P(I_i)$ is precision at i^{th} relevant image. Table III shows mAP results with cosine similarity metric while the outputs generated with the intersection similarity measurement technique is given Table IV. In addition, the *mAP* values of each class are graphically shown in Fig. 7, Fig. 8, Fig. 9 and Fig. 10. Also, the P-R curves of the overall system for the top 20 similar images are shown in Fig. 11. All experiments were repeated with cosine and histogram intersection similarity algorithms for LBG strategy and ODH generated with Kapur's entropy criteria.

TABLE III. *MAP* WITH COSINE SIMILARITY (TOP 20): ODH AND LBG

Categories of Corel1K	CBIR with entropy				CBIR with LBG		
	Number of Clusters				Number of Clusters		
	8	27	64		8	27	64
Africans	0,67	0,69	0,79		0,59	0,60	0,61
Beaches	0,69	0,72	0,73		0,58	0,59	0,60
Buildings	0,63	0,64	0,67		0,58	0,59	0,60
Buses	0,64	0,65	0,68		0,59	0,60	0,61
Dinosaurs	0,95	1,00	1,00		0,94	0,97	0,99
Elephants	0,72	0,74	0,75		0,57	0,58	0,6
Flowers	0,75	0,83	0,83		0,59	0,60	0,63
Horses	0,81	0,89	0,93		0,58	0,59	0,62
Mountains	0,64	0,66	0,67		0,56	0,58	0,58
Foods	0,71	0,76	0,79		0,57	0,57	0,58
Averages	0,72	0,76	0,78		0,62	0,63	0,64

TABLE IV. *MAP* WITH INTERSECTION SIMILARITY (TOP 20): ODH AND LBG

Categories of Corel1K	CBIR with entropy				CBIR with LBG		
	Number of Clusters				Number of Clusters		
	8	27	64		8	27	64
Africans	0,67	0,70	0,81		0,58	0,60	0,61
Beaches	0,69	0,71	0,74		0,58	0,59	0,61
Buildings	0,64	0,66	0,69		0,58	0,58	0,59
Buses	0,67	0,68	0,68		0,58	0,58	0,59
Dinosaurs	0,96	1,00	1,00		0,96	0,98	0,99
Elephants	0,74	0,74	0,74		0,57	0,59	0,61
Flowers	0,75	0,82	0,86		0,6	0,61	0,65
Horses	0,82	0,90	0,93		0,59	0,61	0,63
Mountains	0,65	0,66	0,67		0,58	0,59	0,60
Foods	0,70	0,77	0,82		0,57	0,58	0,58
Averages	0,73	0,76	0,79		0,62	0,63	0,65

Averages rows in Table III and Table IV are important as they represent an overview of the situation in the Corel 1K data set. When these values are examined, it is clear that the ODH algorithm is more successful than the LBG method in both similarity methodologies. Fig. 7, Fig. 8, Fig. 9 and Fig 10 clearly show the achievements.

From the perspective of similarity measurements of CBIR systems, it is clear that the algorithms tested with the histogram intersection similarity measurement technique largely retrieve images that are more relevant than with cosine method. The overall performance of the histogram intersection strategy could also be seen in Fig. 11 where the area under P-R curve is maximum.

Another finding in Table III and Table IV is that when the number of thresholds increases, the mAP value improves as well. While the *mAP* result obtained by Ashraf et al. [44] for the same database was 0.73, in this study, the mAP value has been increased up to 0.78 by cosine similarity metric and 64 clusters. Moreover, it was reached to 0.79 by intersection similarity metric as shown in Table IV.

The performance of all techniques is generally low in Beaches, Buildings and Mountains. Because the images in these categories have a complex color group structure due to the objects they contain. The difference in light intensity due to the daylight or dark shots of the images in these categories is important factors affecting the performance of system. It is clear, however, that feature vectors obtained with entropy in related classes still represent images better. The developments attained by the proposed ODH can be seen in Fig. 7 and Fig. 9. Buses class is also hard one for retrieval according to *mAP* values. The retrieval process with all methods is generally the most successful with Dinosaurs, because the backgrounds of the images in this class are very similar and consist of a single object.

The achievements with Elephants are promising with the developed algorithm. The significant improvements are realized in Flowers and Horses classes. The success of ODH technique could be seen in Fig. 7 and Fig. 9, too. Finally, considerable achievements with Foods class have been realized. Subsequently, the overall P-R curves given in Fig.11 confirm the achievements given in Table III and Table IV.

The developed scheme has capability to improve the performance of CBIR system by increasing the number of thresholds. Consequently, it is observed that the proposed content-based image retrieval algorithm with cluster-based one-dimensional histogram (ODH) is robust scheme. It is also superior to LBG algorithm in both similarity measurements.

The LBG algorithm for color reduction and feature extraction relies on the initial values of clusters and number of iteration. Therefore, it is a kind of user dependent techniques. On the other hand, the proposed entropy based approach does not need any initial values of cluster. Moreover, as entropies of digital image were used to construct feature vectors, the performance of the devised CBIR system has been enhanced.

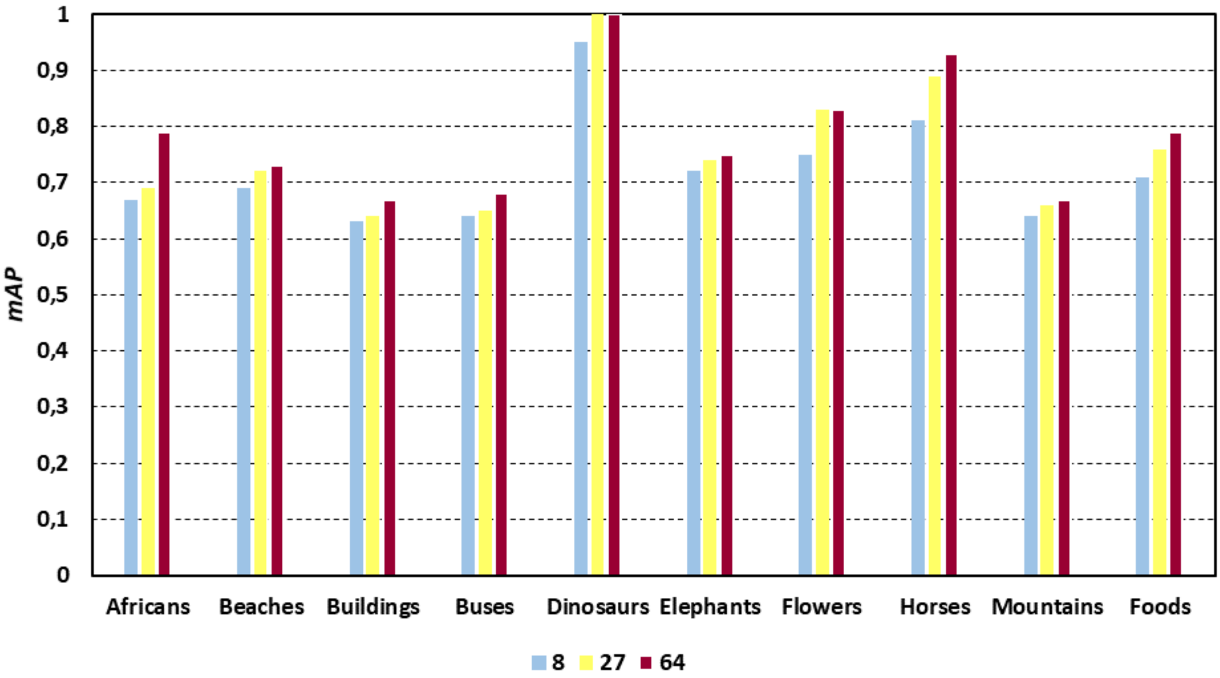

Figure 7. *mAP* with cosine similarity (top 20): ODH

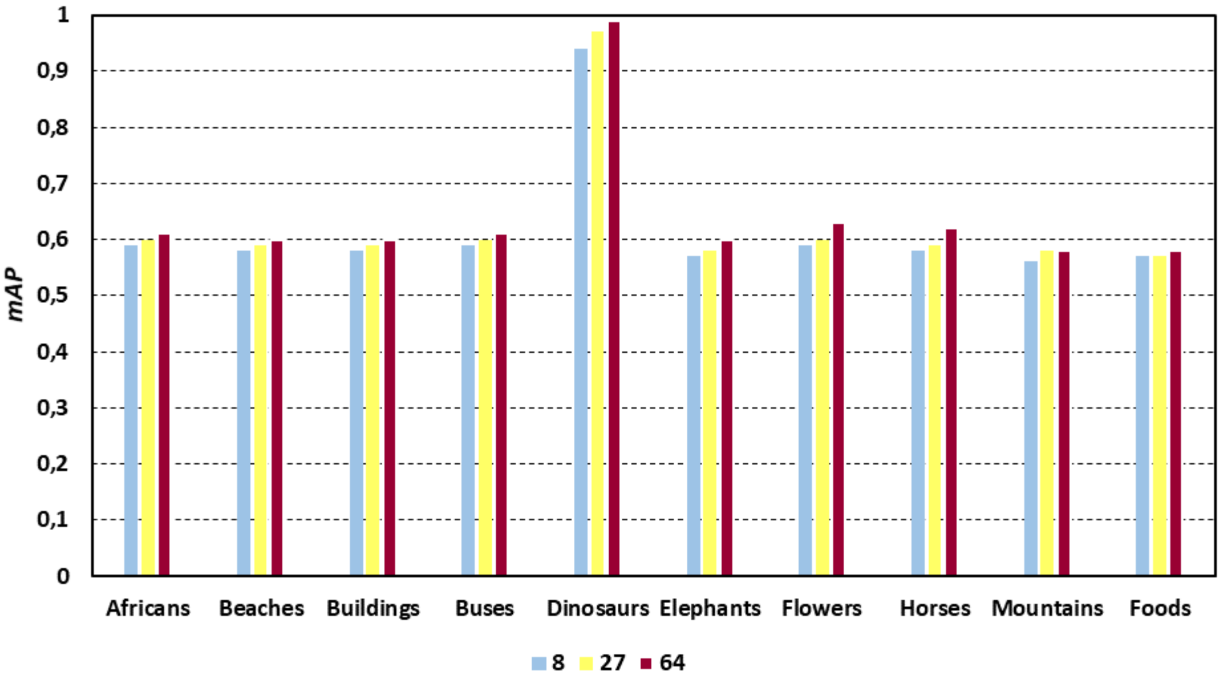

Figure 8. *mAP* with cosine similarity (top 20): LBG

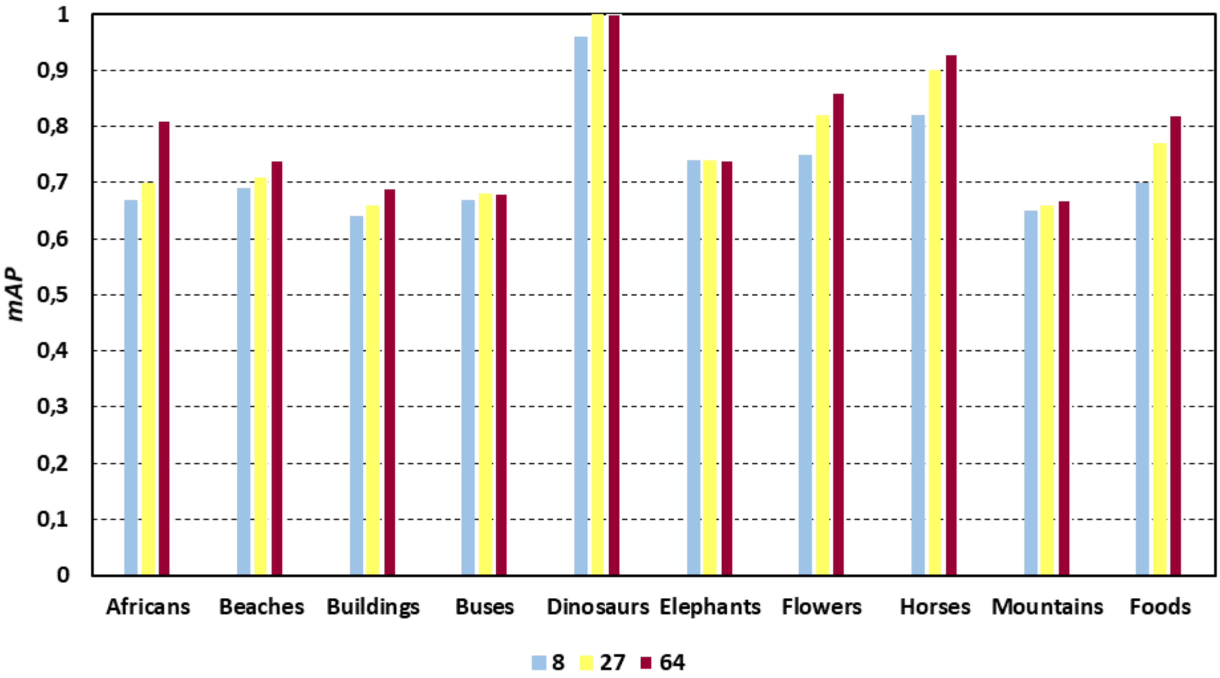

Figure 9. *mAP* with histogram intersection similarity (top 20): ODH

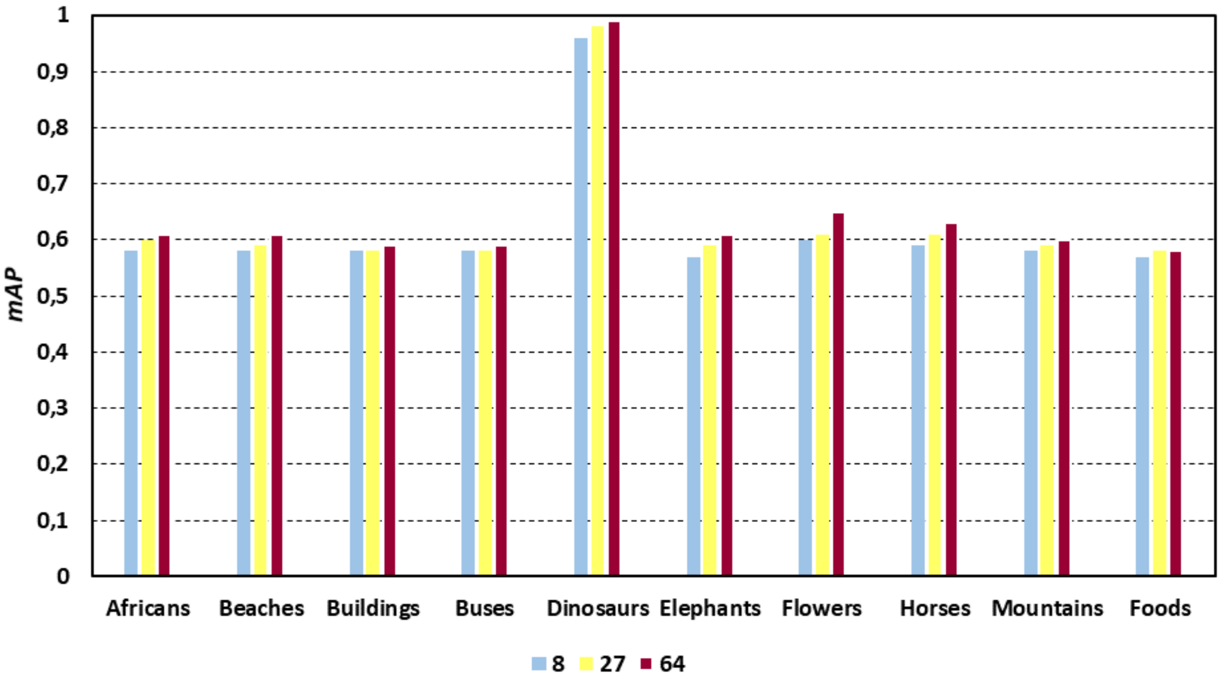

Figure 10. *mAP* with histogram intersection similarity (top 20): LBG

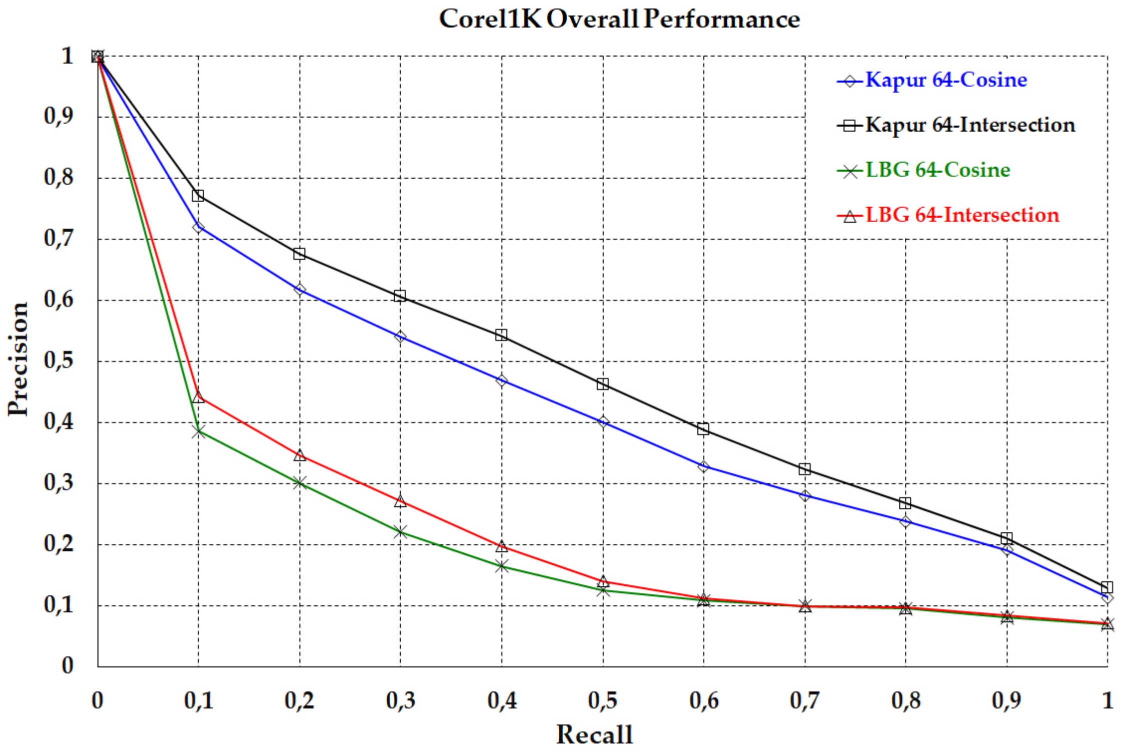

Figure 11. Overall precision-recall curves with different algorithms

V. Conclusion

The CBIR systems are fundamentally based on the feature extraction stage, which converts image information into a set of properties, in other words, into feature vectors. The related feature vectors are crucial for the accurate and fast retrieval systems. Histograms of images are also widely preferred in feature extraction phase of CBIR systems. In this context, it is necessary to combine information obtained from three different channels in the color images. Other solution is to use three dimensional color histograms which have highly computation cost. In this study, a novel image retrieval approach using one-dimensional histogram (ODH) for color images has been proposed. The color images were initially clustered by means of multi-level image thresholding technique. The entropies of histograms were evaluated for threshold calculations. Furthermore, the averages of clusters obtained as results of thresholding process were used for color reduction and to generate the one-dimensional histograms (ODH). The one-dimensional histogram obtained for every color image was employed as a feature vector for content-based image retrieval system. The performance of suggested algorithm was compared with the well-known color quantization algorithm, LBG. The LBG algorithm depends the initial values of cluster centers and it is an iterative procedure whereas the established strategy does not work with iterative manner and require any initial estimates of clusters. Therefore, it produces the same results whenever it runs. The suggested approach is tested with two well-known histogram similarity measurements methods. Finally, it is observed that the use of intersection of histograms in CBIR system produced a more effective result than the cosine similarity metric, and that the entropy based feature vectors and intersection of histograms for similarity measure in CBIR systems are compatible.

References

[1] T. Y. S. Rao and P. C. Reddy, "Content and context based image retrieval classification based on firefly-neural network," Multimedia Tools and Applications, vol. 77, no. 24, pp. 32041-32062, 2018. doi:10.1007/s11042-018-6224-x

[2] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing," IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 276-286, Mar. 2018. doi:10.1109/TCC.2015.2491933

[3] H. Tamura and N. Yokoya, "Image database systems: A survey," Pattern recognition, vol. 17, no. 1, pp. 29-43, 1984. doi:10.1016/0031-3203(84)90033-5

[4] S.-K. Chang and A. Hsu, "Image information systems: where do we go from here?," IEEE transactions on Knowledge and Data Engineering, vol. 4, no. 5, pp. 431-442, 1992. doi:10.1109/69.166986

[5] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," Journal of visual communication and image representation, vol. 10, no. 1, pp. 39-62, 1999. doi:10.1006/jvci.1999.0413

[6] P. Bosilj, E. Aptoula, S. Lefèvre, and E. Kijak, "Retrieval of remote sensing images with pattern spectra descriptors," ISPRS International Journal of Geo-Information, vol. 5, no. 12, pp. 228-243. doi:10.3390/ijgi5120228

[7] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applications-clinical benefits and future directions," International journal of medical informatics, vol. 73, no. 1, pp. 1-23, 2004. doi:10.1016/j.ijmedinf.2003.11.024

[8] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," Pattern recognition, vol. 40, no. 1, pp. 262-282, 2007. doi:10.1016/j.patcog.2006.04.045

[9] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," ACM Computing Surveys (Csur), vol. 40, no. 2, p. 5, 2008. doi:10.1145/1348246.1348248

[10] N. S. Vassilieva, "Content-based image retrieval methods," Programming and Computer Software, vol. 35, no. 3, pp. 158-180, 2009. doi:10.1134/S0361768809030049

[11] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," arXiv preprint arXiv:1706.06064, 2017. https://arxiv.org/abs/1706.06064

[12] N. Ghosh, S. Agrawal, and M. Motwani, "A survey of feature extraction for content-based image retrieval system," in Proceedings

of International Conference on Recent Advancement on Computer and Communication, 2018, pp. 305-313. doi:10.1007/978-981-10-8198-9_32

[13] A. Atto, Y. Berthoumieu, and R. Mégret, "Stochasticity: a feature for the structuring of large and heterogeneous image databases," Entropy, vol. 15, no. 11, pp. 4782-4801, 2013. doi:10.3390/e15114782

[14] H. Zhao, Q. Li, and P. Liu, "Hierarchical geometry verification via maximum entropy saliency in image retrieval," Entropy, vol. 16, no. 7, pp. 3848-3865, 2014. doi:10.3390/e16073848

[15] R. Ashraf, K. Bashir, A. Irtaza, and M. Mahmood, "Content based image retrieval using embedded neural networks with bandletized regions," Entropy, vol. 17, no. 6, pp. 3552-3580, 2015. doi:10.3390/e17063552

[16] X. Lu, J. Wang, X. Li, M. Yang, and X. Zhang, "An Adaptive Weight Method for Image Retrieval Based Multi-Feature Fusion," Entropy, vol. 20, no. 8, p. 577, 2018. doi:10.3390/e20080577

[17] A. Eshkol, M. Grega, M. Leszczuk, and O. Weintraub, "Practical application of near duplicate detection for image database," in International Conference on Multimedia Communications, Services and Security, 2014, pp. 73-82. doi:10.1007/978-3-319-07569-3_6

[18] A. Talib, M. Mahmuddin, H. Husni, and L. E. George, "A weighted dominant color descriptor for content-based image retrieval," Journal of Visual Communication and Image Representation, vol. 24, no. 3, pp. 345-360, 2013. doi:10.1016/j.jvcir.2013.01.007

[19] V. Kubicova and I. Provaznik, "Use of whole genome DNA spectrograms in bacterial classification," Computers in biology and medicine, vol. 69, pp. 298-307, 2016. doi:10.1016/j.compbiomed.2015.04.038

[20] C. Reta, J. A. Cantoral-Ceballos, I. Solis-Moreno, J. A. Gonzalez, R. Alvarez-Vargas, and N. Delgadillo-Checa, "Color uniformity descriptor: An efficient contextual color representation for image indexing and retrieval," Journal of Visual Communication and Image Representation, vol. 54, pp. 39-50, 2018. doi:10.1016/j.jvcir.2018.04.009

[21] J. Jing, Q. Li, P. Li, and L. Zhang, "A new method of printed fabric image retrieval based on color moments and gist feature description," Textile Research Journal, vol. 86, no. 11, pp. 1137-1150, 2016. doi:10.1177/0040517515606378

[22] P. Xu, S. Guo, Q. Miao, B. Li, X. Chen, and D. Fang, "Face detection of golden monkeys via regional color quantization and incremental self-paced curriculum learning," Multimedia Tools and Applications, vol. 77, no. 3, pp. 3143-3170, 2018. doi:10.1007/s11042-017-4984-3

[23] Lu-yu Chen and Z. Chun-yan, "Image retrieval algorithm based on block color histogram and GWLBP," Chinese Journal of Liquid Crystals and Displays, vol. 32, no. 9, pp. 755-763, 2017. doi:10.3788/YJYXS20173209.0755

[24] N. Shrivastava and V. Tyagi, "An efficient technique for retrieval of color images in large databases," Computers & Electrical Engineering, vol. 46, pp. 314-327, 2015. doi:10.1016/j.compeleceng.2014.11.009

[25] N. Varish, J. Pradhan, and A. K. Pal, "Image retrieval based on non-uniform bins of color histogram and dual tree complex wavelet transform," Multimedia Tools and Applications, vol. 76, no. 14, pp. 15885-15921, 2017. doi:10.1007/s11042-016-3882-4

[26] P. Liu, J.-M. Guo, K. Chamnongthai, and H. Prasetyo, "Fusion of color histogram and LBP-based features for texture image retrieval and classification," Information Sciences, vol. 390, pp. 95-111, 2017. doi:10.1016/j.ins.2017.01.025

[27] S. O. Abter and N. A. Abdullah, "An efficient color quantization using color histogram," in 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017, pp. 13-17. doi:10.1109/NTICT.2017.7976153

[28] K. Chiranjeevi and U. R. Jena, "Image compression based on vector quantization using cuckoo search optimization technique," Ain Shams Engineering Journal, vol. 9, no. 4, pp. 1417 - 1431, 2018. doi:10.1016/j.asej.2016.09.009

[29] A. Parsi, A. G. Sorkhi, and M. Zahedi, "Improving the unsupervised LBG clustering algorithm performance in image segmentation using principal component analysis," Signal, Image and Video Processing, vol. 10, no. 2, pp. 301-309, 2016. doi:10.1007/s11760-014-0742-4

[30] P.-Y. Yang, J.-T. Tsai, and J.-H. Chou, "PCA-based fast search method using PCA-LBG-based VQ codebook for codebook search," IEEE Access, vol. 4, pp. 1332-1344, 2016. doi:10.1109/ACCESS.2016.2548664

[31] J.-T. Tsai, P.-Y. Chou, and J.-H. Chou, "Performance comparisons between PCA-EA-LBG and PCA-LBG-EA approaches in VQ codebook generation for image compression," International Journal of Electronics, vol. 102, no. 11, pp. 1831-1851, 2015. doi:10.1080/00207217.2014.996785

[32] J.-T. Tsai and P.-Y. Yang, "PCA-LBG-based algorithms for VQ codebook generation," International Journal of Electronics, vol. 102, no. 4, pp. 529-547, 2015. doi:10.1080/00207217.2014.905994

[33] O. Buyuk and M. L. Arslan, "Combination of Long-Term and Short-Term Features for Age Identification from Voice," Advances in Electrical and Computer Engineering, vol. 18, no. 2, pp. 101-109, 2018. doi:10.4316/AECE.2018.02013

[34] H. Kekre, T. K. Sarode, S. D. Thepade, and S. Sanas, "Image Retrieval Using Texture Features Extracted as Vector Quantization Codebooks Generated Using LBG and Kekre Error Vector Rotation Algorithm," in Technology Systems and Management, Springer, 2011, pp. 207-213. doi:10.1007/978-3-642-20209-4_29

[35] R. M. Gray and D. L. Neuhoff, "Quantization," IEEE transactions on information theory, vol. 44, no. 6, pp. 2325-2383, 1998. doi:10.1109/18.720541

[36] Y. Ueda, T. Koga, N. Suetake, and E. Uchino, "Color quantization method based on principal component analysis and linear discriminant analysis for palette-based image generation," Optical Review, vol. 24, no. 6, pp. 741-756, 2017. doi:10.1007/s10043-017-0376-1

[37] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," IEEE Transactions on communications, vol. 28, no. 1, pp. 84-95, 1980. doi:10.1109/TCOM.1980.1094577

[38] C. Fonseca, F. A. Ferreira, and F. Madeiro, "Vector quantization codebook design based on Fish School Search algorithm," Applied Soft Computing, vol. 73, pp. 958-968, 2018. doi:10.1016/j.asoc.2018.09.025

[39] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," Computer vision, graphics, and image processing, vol. 29, no. 3, pp. 273-285, 1985. doi:10.1016/0734-189X(85)90125-2

[40] M. Kılıçaslan, U. Tanyeri, M. İncetaş, B. Y. Girgin, and R. Demirci, "Eşikleme Tekniklerinin Renk Uzayı Tabanlı Kümeleme Yönteminin Başarısına Etkisi," in 1st International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT 2017), Tokat, Türkiye, 2017, pp. 107-110.

[41] M. Kılıçaslan, U. Tanyeri, and R. Demirci, "Renkli Görüntüler İçin Tek Boyutlu Histogram," Düzce Üniversitesi Bilim ve Teknoloji Dergisi, vol. 6, no. 4, pp. 1094-1107. doi:10.29130/dubited.413822

[42] T. Rahkar Farshi, R. Demirci, and M.-R. Feizi-Derakhshi, "Image clustering with optimization algorithms and color space," Entropy, vol. 20, no. 4, p. 296, 2018. doi:10.3390/e20040296

[43] R. Tapu, B. Mocanu, and T. Zaharia, "Automatic Assistant for Better Mobility and Improved Cognition of Partially Sighted Persons," Advances in Electrical and Computer Engineering, vol. 15, no. 3, pp. 45-53, 2015. doi:10.4316/AECE.2015.03006

[44] R. Ashraf et al., "Content based image retrieval by using color descriptor and discrete wavelet transform," Journal of medical systems, vol. 42, no. 3, p. 44, 2018. doi:10.1007/s10916-017-0880-7