AN EFFICIENT AND FAST IMAGE INDEXING AND SEARCH SYSTEM BASED ON COLOR AND TEXTURE FEATURES

[1]EL AROUSSI EL MEHDI, [2]EL HOUSSIF NOURDDINE, [3]SILKAN HASSAN
[1,2,3] Faculty of Science, University Chouaib Doukkali, Department of Mathematics and Computer Science
Laboratory LAMAPI, El Jadida Morocco.
E-mail: [1] elaroussi.e@ucd.ac.ma, [2] elhoussif.n@ucd.ac.ma, [3] Silkan.h@ucd.ac.ma

ABSTRACT

Content-Based Image Retrieval (CBIR) allows to automatically extracting target images according to objective visual contents of the image itself. Representation of visual features and similarity match are important issues in CBIR. Color, texture and shape information have been the primitive image descriptors in content-based image retrieval systems. This paper presents a fast and efficient image indexing and search system based on color and texture features. The color features are represented by combines 2-D histogram and statistical moments and texture features are represented by 2-D Localized SDFT that uses the Gaussian kernel to offer the spatial localization ability. 2-D SDFT is expected to provide more useful information. It is observed that color features in combination with the texture features derived from the brightness component provide approximately similar results when color features are combined with the texture features using all three components of color, but with much less processing time. The detailed experimental analysis is carried out using precision and recall on two datasets: Corel-DB, Coil-100. The time analysis is also performed to compare processing speeds of the proposed method with the existing similar best.

Keywords: *CBIR, Color histogram, Texture feature, statistical moments, 2-D Localized SDFT.*

1. INTRODUCTION

In the beginning of this decade, we witnessed a digital revolution in imaging, not least for applications targeting the mass market. The use of digital cameras increased dramatically and thereby the number of digital images. Many of us have private photo collections containing thousands of digital images. Naturally, we share images with each other, and also publish them for instance on the Internet. Those image collections are important contributors to the public domain of the Internet, nowadays containing several billion images. To illustrate the size of this domain, or images on the Internet might be the most obvious example, but the use of digital imaging has spread to many application areas. Newspapers, image providers, and other companies in the graphic design industry are now using digital images in their workflow and databases. Also, modern hospitals are good examples, where a large number of medical images are managed and stored in digital systems every day and the security industry, where surveillance cameras digital formats.

Some image collections are highly organized with keywords or other types of labels, making text-based search efficient for finding a specific image, or images with a particular content. However, many image collections are poorly labeled, or not labeled at all. Consequently, with the number of images increasing, it is necessary to find other ways of searching for images. As an alternative to text-based search, we can think of tools that can" look into images", and retrieve images, or organize large image collections, based on the visual image content. The research area based on this idea is called Content Based Image Retrieval (CBIR). This research area has matured over the years, and nowadays numerous parts of CBIR have found their way into commercial implementations.

Most of the early works on CBIR have focused on a single feature set from among the various features of color, texture, and shape. Often, it is difficult to obtain satisfactory retrieval performance using a single feature. The color feature is the most widely used feature among the various visual features. Therefore, when more than one feature is used, normally color is always combined with other

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

features. The role of color, texture, and shape has been discussed extensively by [1–5]. There are two major problems associated with combining different features. First, the size of the combined feature vector is large as compared to the size of individual feature vectors. This makes the retrieval process slow where accurate and fast retrieval results are required. This factor is quite relevant when the image databases are quite large. Often, the users do not prefer long response time even though such combination of features may yield better retrieval accuracy. Second, combining features from different modalities is not an easy task. If not done carefully, combining features may turn out to provide low retrieval rate than the individual features [6].

In this paper, we analyze retrieval performance using color and texture features. The analysis is performed in four color spaces: RGB, HSV, L*a*b*, and YCbCr. It is applied to two databases Coil-100 and Corel-DB. The color histogram is used for color features and 2-D Localized SDFT features are used for analyzing texture features. Keeping in view its superior performance, the simplicity of implementation and computational efficiency, we select 2-D Localized SDFT texture features to fuse with color features. In order to reduce the size of texture features, the results of texture feature on the brightness component of a color space are compared with the combined features of the three components of the color. It is shown that the brightness component provides satisfactory retrieval results, thus eliminating the need of high dimensionality of texture features of color images. The results of the proposed approach are compared with a recently developed state-of-the-art CDH technique [5] and approach [25] and also the approach [30] it is shown that the proposed technique provides comparable or better image retrieval results at a much faster rate. The paper is developed as follows: section 2 relayed the literature review pertaining to the recent researches in indexation and Image Content. Section 3 discusses the descriptor under study. Section 4 deals with the results of our experiments; and section 5 draws the conclusions of the proposed approach.

2. RELATED WORK

Manuscripts Some of the significant techniques are discussed in this section. A review of various ROI image retrieval techniques is given in [7]. Wang et al. [8] have proposed an image retrieval scheme combining color feature like the dominant color of the region, texture feature like steerable filter and shape feature based on the pseudo-Zernike moment. In Blobworld [9], the quadratic form distance is used to match two color histograms. The distance between two texture descriptors is the Euclidean distance between their coordinates in representation space. The distance between centroids is the Euclidean distance. The distances are combined into a single final distance. In KIWI [10] (Key-points Indexing Web Interface), the color feature vectors are compared with the Euclidean distance and 2D histograms are compared using the Bhattacharyya distance. After a normalization of the distribution of distance values from the individual features, the similarity values are sorted and then averaged. In Metaseek [11], color and texture are extracted locally by Metaseek for the clustering. The color similarity is computed by calculating the color histogram of an image. The texture is computed by measuring the coarseness, contrast, and presence/ absence of directionality of an image. The distance between two feature vectors is the Euclidean distance. Lu and Chang [12] used two different measurements of the global features and the local feature to evaluate the similarity between the two images. For the global color features, the scheme uses Euclidean distance to calculate the similarity.

The texture features were combined with color features by Chun et al. [1] for color image retrieval and obtained very good results. Liapis and Tziritas [2] used the color histogram in L*a*b*color space and variance extracted from discrete wavelet transform as texture features. Deselaers et al. [3] have conducted experimental performance analysis using color histograms and various features involving texture and shape. The color histogram is observed to be an effective image descriptor and it is suggested as a baseline descriptor for many applications. According to their observations, none of the shape or texture features were found to be as effective as the color feature. They have also performed correlation analysis for combining various features and suggested some directions for the future work. The color correlogram is also a good example which combines color features and spatial distribution of pixels. Recently, color, texture and shape features have been studied extensively by Liu et al.[4,5]. An approach combining the features of color, texture, and shape has been discussed in [5] which is based on color difference histogram (CDH) in L*a*b* color space. Chawki et al. [30] have proposed to use GF with Bidimensional High-Resolution Spectral Analysis 2-D ESPRIT (Estimation of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Signal Parameters via Rotational Invariance Techniques) method to extract image characteristics and for constructing a new descriptor vector. Yu et al [13] presented a color texture moments method by using local Fourier transform based on HSV color space and derived eight characteristic maps for describing co-occurrence relations of image pixels in each color channel. CIE Lab color space has been employed by researchers because of its perceptual uniformity: Liapis and Tziritas [14] extracted the texture features in CIE Lab color space; Guang-Hai and Jing-Yu [15] proposed color difference histograms which count the color difference between two pixels in CIE Lab space. Color feature combing other features such as texture feature and shape information can achieve higher retrieval efficiency. Hiremath and Pujari [16] served the color moments and the moments on the Gabor filter responses as local descriptors of color and texture respectively; the shape information is captured in terms of edge images. Similarly, by considering the texture feature and color feature in color space, Cong Bai et al [17] developed the histogram of feature vectors which are constructed from the sub-bands of the wavelet transform. In most of the cases, image representation methods extract the color features at each color channels independently. In fact, there exist dependencies caused by linear transform in the color space. In recent years, copula has been used to capture the color dependence and achieved success [18, 19]. In this work, we propose a texture retrieval method with copula and the Gabor wavelets [20] which consist of Gabor filters. Gabor filter can well model the cells of the visual cortex of human; therefore, it is useful to generate the textons of the image [21, 22]. Our method not only considers the RGB color dependence but also captures the dependencies in the subbands of Gabor wavelets by using copula which separates a dependence structure among variables from its marginal distributions. According to structure characteristic of Gabor wavelets subbands in the RGB color space, we develop four copula schemes for the representation of the RGB texture image. Regarding the feature matching, in this work, Kullback-Leibler Distance (KLD) is used. For the sake of computational efficiency, we derive the closed-form KLD between two copula models by summing the closed-form KLD of marginal models and the closed-form KLD of copula functions. Compared with [18], which employs the DTCWT to decompose image and ML (Maximum Likelihood) to retrieval images, our method uses Gabor wavelets to decompose images and uses KLD as the similarity of copula models. Our previous work also uses Gabor wavelets to extract a rotation-invariant feature of the gray scale image [23] by taking into account the scale dependence of Gabor wavelets. The color information of the image is also used in Elasnaoui et al. [24] the proposed approach is based on the intersection of 2-D histograms in HSV space. The used histogram is based not only on the intensity of pixels but also on a 3x3 window. This approach overcomes the drawback of the classical histogram, which ignores the spatial distribution of pixels in the image. To escape the effects of the discretization of the color space, which is intrinsic to the use of histograms [24] another approach has been successfully used by Elasnaoui et al. [25] This approach combines 2-D histogram and statistical moments in the HSV color space and it is applied to different real images to demonstrate the performance of the algorithm for color image retrieval.

3. FEATURE EXTRACTION

3.1 Color feature extraction

The color feature is one of the most widely used visual features in image retrieval. Typically, the color of an image is represented through some color model. There exist various color models to describe color information. A color model is specified in terms of 3-D coordinate system and a subspace within that system where each color is represented by a single point. The more commonly used color models are RGB (red, green, blue) and HSV (hue, saturation, value). Thus the color content is characterized by 3-channels from some color model. One representation of the color content of the image is by using the color histogram. Statistically, it denotes the joint probability of the intensities of the three-color channels. The color is perceived by humans as a combination of three-color stimuli: Red, Green, and Blue, which forms a color space. RGB colors are called primary colors and are additive. By varying their combinations, other colors can be obtained. The representation of the HSV space is derived from the RGB space cube, with the main diagonal of the RGB model, as the vertical axis in HSV.

3.1.1 Color Histogram

Each image in the database is computed to obtain the color histogram, which shows the proportion of pixels of each color within the image. The color histogram of each image is then stored in the database. When the user does the search by specifying the query image, the system registers the proportion of each color of the query image and

Journal of Theoretical and Applied Information Technology
15th June 2018. Vol.96. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

goes through all images in the database to find those whose color histograms match those of the query most closely. The color histograms are used to represent the color distribution in an image. Mainly, the color histogram approach counts the number of occurrences of each unique color on a sample image. Since an image is composed of pixels and each pixel has a color, the color histogram of an image can be computed easily by visiting every pixel once. By examining the color histogram of an image, the colors existing on the image can be identified with their corresponding areas as the number of pixels. Histogram search characterizes an image by its color distribution or histogram. Euclidian histogram distances have been used to define the similarity of two color histogram representations. It is the most used descriptor in image retrieval. The color histogram is easy to compute, simple and effective in characterizing the global and the local distribution of colors in an image. The color histogram extraction algorithm uses three steps: partition of the color space into cells, an association of each cell to a histogram bin, and counting of the number of image pixels of each cell and storing this count in the corresponding histogram bin.

3.1.2. Color Moments

Color moments have been successfully used in several retrieval systems. This approach involves calculating the mean, the variance and the third moment for each color channel, for providing a unique number used to index. Color moments have been proved efficient in representing color distributions of images. They are defined as:

Mean: $\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - \mu_i)^2}$

Standard Deviation: $\mu_i = \frac{1}{N}\sum_{j=1}^{N} P_{ij}$

Skewness: $s_i = (\sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - \mu_i)^3})$

3.2 Texture feature extraction

The texture is a feature that is quite difficult to describe, and subjected to the difference of human perception. The texture is the property of an image, characterized by the existence of basic primitives whose spatial distribution creates some visual patterns generally, texture representation methods are classified into two main categories: structural and statistical. Structural methods, including morphological operator and adjacency graph. Statistical methods, including Tamura feature, shift-invariant principal component analysis (SPCA), and multi-resolution filtering techniques such as Gabor and wavelet transform, define texture by the statistical distribution of the image intensity.

The Gabor filter offers the good spatial localization thanks to the Gaussian kernel that determines an weight based on a spatial distance. Inspired by this, we present new methods that efficiently compute the 2-D localized SDFT using the proposed kernel decomposition technique. Different from the existing 2-D SDFT approaches [26]–[27] using the box kernel, we use the Gaussian kernel when computing the DFT at the sliding window as in the Gabor filter. It should be noted that applying the existing 2-D SDFT approaches [26]–[27] are infeasible in the case of calculating the DFT outputs with the Gaussian kernel.

A. Kernel decomposition in 2-D localized SDFT

When the sliding window of $M \times M$ is used, we set the standard deviation σ of the Gaussian kernel by considering a cut-off range, e.g., $\lfloor M/2 \rfloor = 3\sigma$. We denote $F_{u,v}(x,y)$ by the $(u,v)^{th}$ bin of the $M \times M$ DFT at (x; y) of 2-D image f. The 2-D localized SDFT with the Gaussian kernel can be written as $F_{u,v}(x,y) = \sum_{n,m} f(m,n) C_{u,v}(\hat{x} - m, \hat{y} - n) G_\sigma(x-m, y-n)$ (1)

Where $\hat{x} = x - \frac{M}{2}$ and $\hat{y} = y - \frac{M}{2}$. For u, v = 0... M – 1 the complex exponential function $C_{u,v}(m,n)$ at the $(u,v)^{th}$ frequency is defined as

$C_{u,v}(x,y) = e^{i(\omega_0 ux + \omega_0 vy)}$, (2)

Where $\omega_0 = \frac{2\pi}{M}$. Note that in (1) and (2), slightly different notations than the conventional SDFT methods [26]–[27] are used to keep them consistent with the Gabor filter.

When $G_\sigma(x,y) = 1$, (1) becomes identical to that of the conventional SDFT methods [27]–[29]. The Gaussian window of M × M is used here, but the 2-D localized SDFT using $M_y \times M_x$ window ($M_y \neq M_x$) is also easily derived. Using the separable

Journal of Theoretical and Applied Information Technology
15th June 2018. Vol.96. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

property of $G_\sigma(x,y) = S_\sigma(x)S_\sigma(y)$ and $C_{u,v}(x,y) = H_u(x)V_v(y)$, (17) can be written as

$$J_u(x,y) = \sum_m f(m,y)H_u(\hat{x}-m)\, S_\sigma(x-m),$$

$$F_{u,v}(x,y) = \sum_n J_u(x,n)V_v(\hat{y}-n)S_\sigma(y,n),$$

Using the kernel decomposition, the 1-D horizontal localized SDFT is performed as follows:

$\mathrm{R}\{J_u(x)\} = \cos(\omega_0 u\hat{x})\left(f_c(m) - S_\sigma(x,m)\right) - \sin(\omega_0 u\hat{x}) \sum_m f_c\,(m)S_\sigma(x-m)$ (3)

$\mathrm{I}\{J_u(x)\} = -\cos(\omega_0 u\hat{x}) \sum_m f_s(m)S_\sigma(x-m) + \sin(\omega_0 u\hat{x}) \sum_m f_c(m)S_\sigma(x-m)$ (4)

where $f_c(m) = f(m)\cos(\omega_0 um)$,$f_s(m) = f(m)\sin(\omega_0 um)$. The vertical 1-D localized SDFT is performed similar to the Gabor filter:

$\mathrm{R}\{F_{u,v}(x,y)\} = \cos(\omega_0 v\hat{y})\left(f'_{cr}(x,y) - f'_{si}(x,y)\right) - \sin(\omega_0 v\hat{y})(f'_{sr}(x,y) + f'_{ci}(x,y))$ (5)

$\mathrm{I}\{F_{u,v}(x,y)\} = \sin(\omega_0 u\hat{y})\left(f'_{cr}(x,y) - f'_{si}(x,y)\right) - \cos(\omega_0 v\hat{y})(f'_{sr}(x,y) + f'_{ci}(x,y))$ (6)

B. Exploring Computational Redundancy on (u; v)

The 2-D localized SDFT requires computing a set of DFT outputs for $u,v = 0,\dots,M-1$, similar to the 2-D complex Gabor filter bank. Considering the conjugate symmetry property of the DFT ($F_{M-u,M-v} = F^*_{u,v}$), we compute the DFT outputs $F_{u,v}$ only $u = 0,\dots,M-1$ and $v = 0,\dots,\left\lfloor\frac{M}{2}\right\rfloor$, and then simply compute remaining DFT outputs (pour $u = 0,\dots,M-1$et $v = \left\lfloor\frac{M}{2}\right\rfloor + 1,\dots,M-1$) by using the complex conjugation. Thus, we focus on the computation of the 2-D SDFT for $u = 0,\dots,M-1$ and $v = 0,\dots,\left\lfloor\frac{M}{2}\right\rfloor$, Let us consider how to compute $F_{M-u,v}$ sing intermediate results of $F_{u,v}$.. Similar to the Gabor filter bank, the 1-D DFT J_{M-u} is complex conjugate to J_u as follows:

$$J_{M-u}(x,y) = \sum_m f(m,y)H_{M-u}(\hat{x}-m)\, S_\sigma(x-m),$$

$$= \sum_m f(m,y)H^*_u(\hat{x}-m)\, S_\sigma(x-m),$$

$$= J^*_u(x,y)$$

(7)

The 1-D vertical SDFT result $F_{M-u,v}$ is then obtained as

$F_{M-u,v}(x,y) = \sum_n J^*_u\,(x,y)V_v(\hat{y}-n)S_\sigma(y-n)$

(8)

Algorithm1 shows the overall process of computing the 2-D localized SDFT. Here, we explain the method with a nonsquare window of $M_y \times M_x$ ($M_y \geq M_x$)for a generalized description. This can be simply modified when$M_y < M_x$. Note that when$M_y \geq M_x$, a horizontal filtering (line 4-9 of Algorithm 1) should be performed first and vice versa in order to reduce the runtime. This filtering order does not affect the computational complexity of the 1-D SDFT in line 13- 18. In contrast, the 1-D SDFT in line 4 - 9 is affected when$M_y \neq M_x$, and thus we should perform the 1-D filtering for $u = 0,\dots,\left\lfloor\frac{M_x}{2}\right\rfloor$ On the horizontal direction in line 4 - 9 if$M_y \geq M_x$.

To obtain $M_y \times M_x$DFT outputs at the sliding window of the input image *f* in Algorithm 1, we first obtain $J_u(x,y)$for $u = 0,\dots,\left\lfloor\frac{M_x}{2}\right\rfloor$ by using (19), and then simply calculate $J_u(x,y)$ for $u = \left\lfloor\frac{M_x}{2}\right\rfloor + 1,\dots M_x - 1$ using (7). $J_u(x,y)$computed once using the horizontal filtering can be used to obtain $F_{u,v}(x,y)$ by performing the 1-D vertical filtering. Thus, the horizontal filtering $J_u(x,y)$ is performed only for $u = 0,\dots,\left\lfloor\frac{M_x}{2}\right\rfloor$while the vertical filtering v (x ; y) is done for $u = 0,\dots,M_x-1$ and $v = 0,\dots,\left\lfloor\frac{M_x}{2}\right\rfloor$.

1: Input: input image$f(H \times W)$, scale σ, kernel size $M_y \times M_x(M_y \geq M_x)$

2: Output: SDFT outputs at $u = 0,\dots,M_x-1$ and $v = 0,\dots,M_y-1$

3: $M_{xh} = \left\lfloor\frac{M_x}{2}\right\rfloor$, $M_{yh} = \left\lfloor\frac{M_y}{2}\right\rfloor$

4: for $u = 0,\dots,M_{xh}$ do

5: for $y = 1,\dots,H$ do *1-D horizontal SDFT

6: Perform 1-D Gaussian filtering of f_c, f_s in (3) and (4).

7: Compute $J_u(x,y)$ for all x.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
    end for
end for
for u = M_xh + 1, ....., M_x − 1 do
        J_{M_x−u}(x, y) = J*_u(x, y) ∀(x, y).
end for
for u = 0, ......, M_x − 1 , v = 0, ....., M_yh do
    for x = 1, ....., W do          1-D vertical SDFT
        Perform 1-D Gaussian filtering of f_cr + f_si , f_sr − f_ci in (5) and (6)
        Compute F_{u,v}(x, y) ∀y
    end for
end for
for u = 0, ......, M_x − 1 , v = M_yh + 1, ......, M_y − 1 do
    F_{u,v}(x, y) = F*_{M_x−u, M_y−v}(x, y)
end for
```

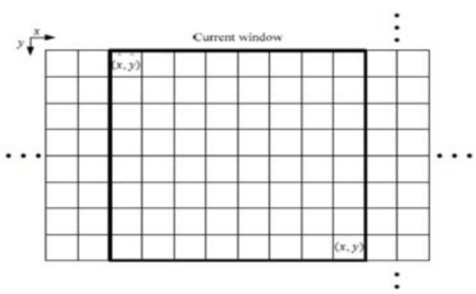

Figure 1. Graphical explanation of current window of size 8×8, where the bold lines denote the current sliding window.

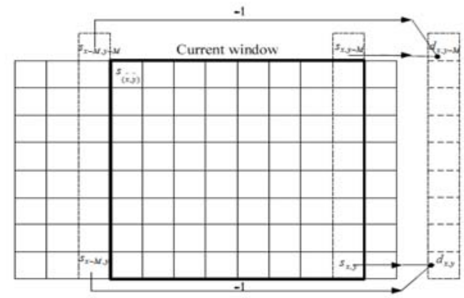

Figure 2.Graphcal explanation of $d_{x,y}$, for sliding window of size 8×8, where the bold lines denote the current sliding

3.3 Image retrieval technique

In the below figure3, the way to create the final descriptor is performed in a distributed computing. Indeed, the input image will be indexed as follows: we first create two processes. The first one calculates histogram 2-D with statistical moments, while the second process calculates 2-D Localized SDFT to create texture descriptor. These two processes work in parallel, and finally, we combine the results to have the final descriptor. This technique accelerates our algorithm according to execution time.

The 2-D-Moments-2D Localized SDFT feature vector is obtained by combining the 2-D-Moments feature and texture feature vectors by normalizing the 2-D-Moments feature and texture feature respectively by z-score normalization. The final descriptor is given by: V = {f2D−Moment} ∪ {2D Localized SDFT}

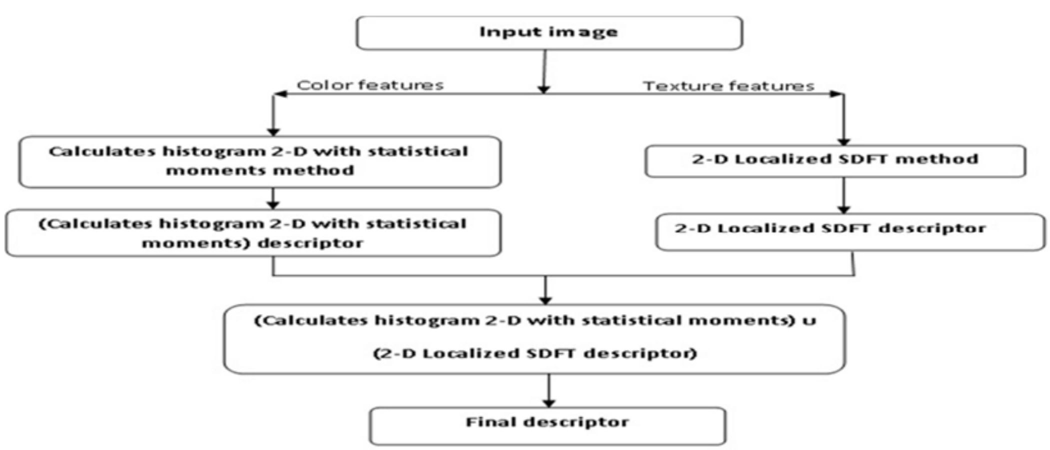

Figure3. Architecture used.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.4 Similarity Computation

Since each distance between features is calculated using a different standard, prior to combining these features, we normalize the distances using Iqbal's method [28].This method ensures that the distance is normalized between 0 and 1. After the normalized distances, we use a linear combination of distances.

$D(Q, I) = w(2D{-}Moment) * D(2D{-}Moment)(Q, I) + w(2D\ Localized\ SDFT) + D(2D\ Localized\ SDFT)(Q, I)$

Where w (2D−Moment and) w (2D Localized SDFT) take a value between 0 and 1.

4. RETRIEVAL EXPERIMENTS

The results and discussion of the proposed content-based image retrieval system are given in this section. The proposed content-based image retrieval system has been implemented in Eclipse JEE Mars (JAVA) under Microsoft Windows 7 environment on an Intel (R) Core (TM) i3 with 2.50 GHz CPU and 4 GB of RAM and the performance of the proposed system is analyzed using the evaluation metrics including precision and recall. Furthermore, two datasets have been used to perform the proposed method.

4.1 Image retrieval

To check the retrieval performance and robustness of the proposed method, several experiments are conducted on Coil-100 and Corel-DB database. We display the best 10 similar images retrieved to prove the performances of our proposed algorithm. The results are presented separately (Figures 4- 5).

Figure 4 Example of a query image and similarity search results in the Coil-100 database and the RGB color space

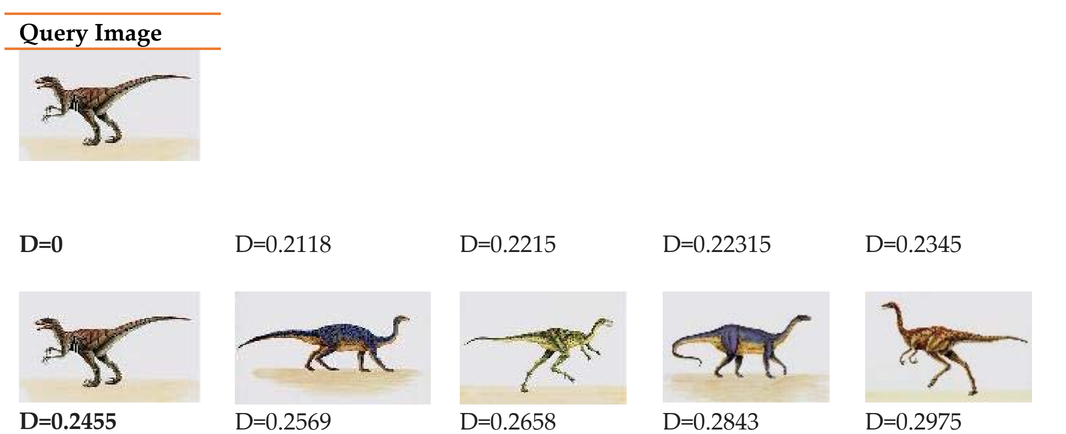

Figure 5. Example Of A Query Image And Similarity Search Results In The Corel-DB Database And The RGB Color Space

4.2 Experimental results

The average precision and recall obtained by the various approaches in different color spaces on Coil-100 dataset are shown in Table 1.

*Table 1: Average precision (%) obtained by various approaches in different color spaces (HSV, RGB, L*a*b*, YCbCr) on Coil-100 dataset*

Approaches	Precision(%)			
	RVB	HSV	L*a*b*	YCbCr
Approach [25]	75	78	62.5	64
Approach [30]	64.35	69.5	56	59
Approach [5]	78.4	82.5	67	70.9
Proposed method	81	86	70	72.5

The average is taken for N = 1–10.The results of the first two approaches show the retrieval performance when the color and texture are used in isolation. It is shown that the color features more effective than the texture features across all color spaces. For the Coil-100 dataset, the approach who combines 2-D histogram and statistical moments (color features) provide the best value of average precision in HSV color space which is 78%. This is followed by the RGB color space which provides the average precision value 75%. The performances of L*a*b* and YCbCr are lower with a value of 62.5% for L*a*b* and 64% for YCbCr. For the second approach (texture features) L*a*b* provides value 56%. This value is lower for the HSV and RGB color spaces, which is 64.35% both for RGB and 69.5% for HSV. When color features are combined with texture features of all components of color space, then again the HSV color space provides the best results of 82.5% for precision as shown in the third row of Table 1. The next best result is obtained for the RGB color space which yields the value 78.4%, follow L*a*b* color space which yields the value 76% and YCbCr color space which yields the value 72.5%. For the proposed method the values of average precision in HSV color space which is 86%. This is followed by the RGB color space which provides the average precision value 81%. The performances of L*a*b* and YCbCr are lower with a value of 70% for L*a*b* and 72.5% for YCbCr. The results of all approaches on large data set Corel-DB are shown in Table 2.

*Table2: Average precision (%) obtained by various approaches in different color spaces (HSV, RGB, L*a*b*, YCbCr) Corel-DB*

Approaches	Précision(%)			
	RGB	HSV	L*a*b*	YCbCr
Approach [25]	60.5	64.55	50	52.6
Approach [30]	49	56.2	41	39.82
Approach [5]	62.9	67.3	52	54.6
Proposed method	66.5	70	55	58

The qualitative behavior of these results is the same as obtained for COIL-100 shown in Table 1. For approach who combines 2-D histogram and statistical moments the best value of average precision in HSV color space which is 64.55%. This is followed by the RGB color space which provides the average precision value 60.5%. For the approach (texture features) provide the value of average precision in HSV color space which is 56.2%.and 49% for RGB color space. For the third approach the best result is obtained for the HSV color space which yields the value 67.5%, follow RVB color space which yields the value 62.9%. Proposed method and approach the precision value is 70% in HSV and 66.5% in RGB color space. This shows that the proposed method provides better results than other methods mentioned above. Thus, it is clear from these tables that the overall best performance is provided in the HSV color space followed by the RGB space. However, there is no significant difference in performance across the various color spaces. We further plot the average precision graphs shown in Figs. 6 and 7, for Coil-100, Corel-DB, respectively, using HSV color space. Proposed method provide the overall best results among the four methods .the results

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

obtained indicate that the proposed approach might be considered as a solution for the development of visual information retrieval. The proposed method provides the overall best results among the four methods with less processing time. We perform the time analysis in Section 4.3

4.3 Etude du temps CPU

Experiments implemented in Eclipse JEE Mars (JAVA) under a Microsoft Windows 7 environment on an Intel (R) Core (TM) i3 with 2.50 GHz CPU and 4 GB of RAM.

The time taken by various methods for indexing and image search is shown in Table 3 for Coil-100 and Corel-DB datasets.

The time taken for feature extraction does not depend on image contents; it depends on the size of the image. It is shown in the table that the feature extraction time the approach [25] who combines 2-D histogram and statistical moments (color features), 0.065 s, is the least and the time for approach [30] 0.195 s. The time taken for the proposed approach is 0.16 s. This is less than the approach [5] which takes 0.25 s. The retrieval times are shown in columns three and four for Coil-100 and Corel-DB datasets. Again, the image retrieval time on COIL-100 data set for the proposed approach is 0.63 s, which is less than the time taken by Method approach [5] which is 1.575 s, which provides the best results. The retrieval time taken on Corel-DB by the proposed method and Method approach [5] is 0.96 s and 0.1.575 s, respectively, which show that the proposed method provides faster image retrieval. The total time is taken for feature extraction and image retrieval for the proposed method which is 1.281 s and 1.9 s for Coil-100 and Corel-DB are much less than that are required for method approach [5] which is 2.073 s for Coil-100 and 3.114 s for Corel-DB, Thus, the proposed method is faster than the two approaches whose retrieval performances are comparable to the performance of the proposed approach

Table3: Time Took (Seconds) By Various Methods For Image Search

Approachs	Time taken for feature extraction (s)	Retrieval time(s)		Total time(s)	
		Coil-100	Corel-DB	Coil-100	Corel-DB
Approach [30]	0.065	0.352	0.528	0.445	0.665
Approach [25]	0.195	0.973	1.455	2.073	2.71
Approach [5]	0.250	1.055	1.575	2.102	3.114
Proposed method	0.16	0.63	0.96	1.281	1.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

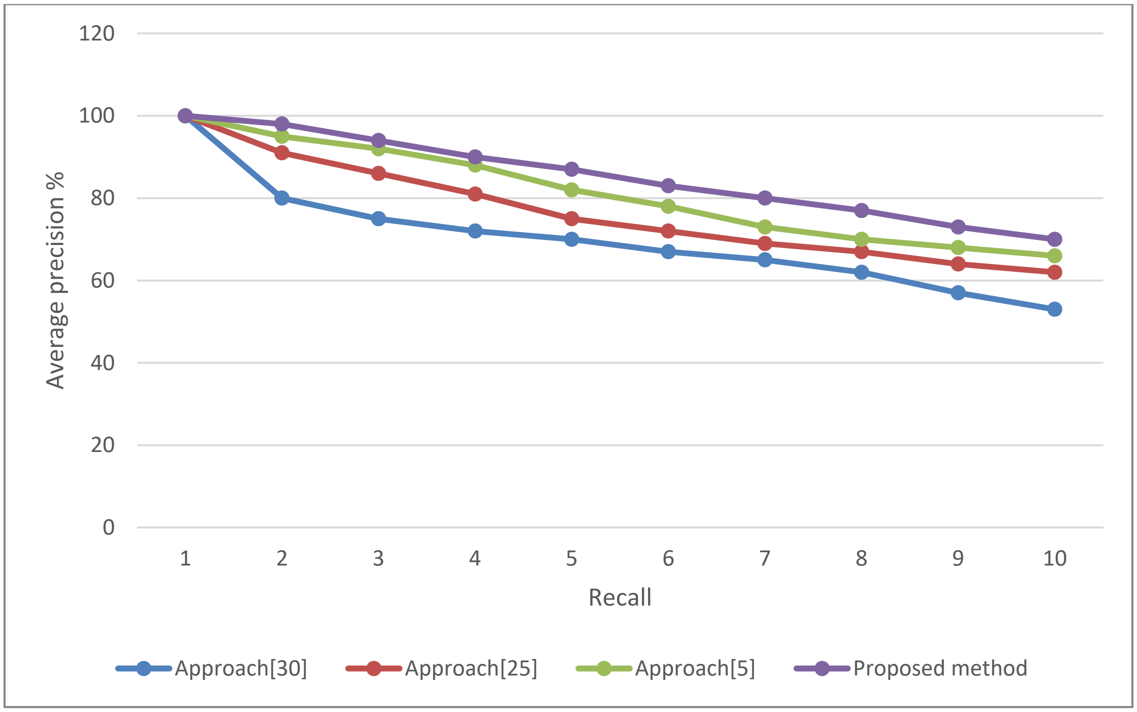

Figure 6: The Average Precision And The Recall Obtained By Some Approaches In The HSV Color Space On The Coil100

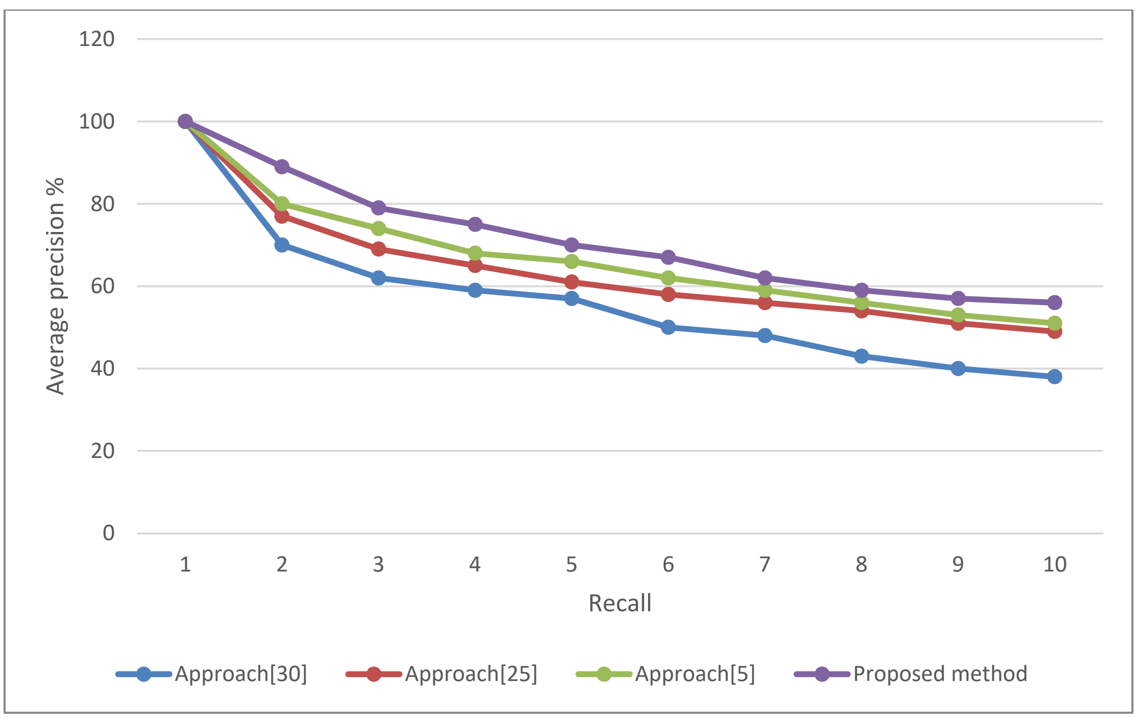

Figure 7: The Average Precision And The Recall Obtained By Some Approaches In The HSV Color Space On The Corel-DB

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSIONS

In this paper, we have proposed a fast and efficient color image retrieval system, by combining color and texture feature for similarity search in two databases. Both types of features are simple to drive and very effective in representing color and texture features. Experimental results based on the vector selectivity and the elapsed time are positive and promising and can improve query performance effectively. It is shown that the brightness component is quite effective when color features are combined with the texture features. Among the various color models, the HSV color model yields overall best performance followed by the RGB color space. In the present analysis, we have used non-training based classifiers because of their simplicity and speed efficiency. In our future work, we shall study the use of training based classifiers such as SVM with the standard kernel functions and v2 and square-chord based kernel functions. Also, features obtained from deep learning with convolutional neural networks (CNN) will be important candidates in our future Works on image retrieved.

REFERENCES:

[1]. Y.D. Chun, N.C. Kim, I.H. Jang, Content-based image retrieval using multi resolution color and texture features, IEEE Trans. Multimedia 10 (6) (2008) 1073–1084.

[2]. S. Liapis, G. Tziritas, Color and texture image retrieval using chromaticity histograms and wavelet frames, IEEE Trans. Multimedia 6 (5) (2004) 676–686.

[3]. T. Deselaers, D. Keysers, H. Ney, Features for image retrieval: an experimental comparison, Inf. Retrieval 11 (2) (2008) 77–107.

[4]. G.-H. Liu, L. Zhang, Y.-K. Hou, Z.-Y. Li, J.-Y. Yang, Image retrieval based on multi text on the histogram, Pattern Recogn. 43 (2010) 2380–2389.

[5]. G.-H. Liu, J.-Y. Yang, Content-based image retrieval using color difference histogram, Pattern Recogn. 46 (2013) 188–198.

[6]. A.K. Jain, K. Nandakumar, Ross, A Score normalization in multimodal biometric systems, Pattern Recogn. 28 (2005) 2270–2285.

[7]. Shrivastava Nishant, Tyagi Vipin. A review of ROI image retrieval techniques. Adv Intel Syst Comput 2015;328:509–20.

[8]. Wang Xiang-Yang, Yu Yong-Jian, Yang Hong-Ying. An effective image retrieval scheme using color, texture & shape features. Comput Stand Interf 2011; 33:59–68.

[9]. Carson Chad, Thomas Megan, Belongie Serge, Hellerstein Joseph M, Malik Jitendra. Blobworld: image segmentation using expectation-maximization and its application to image querying. IEEE Trans Pattern Anal Mach Intell 2002; 24(8):1026–38.

[10]. Loupias E, Bras S. Key point-based indexing for pre-attentive similarities: the kiwi system. Pattern Anal Appl 2001; 4(2/3):200–14.

[11]. Benitez AB, Beigi M, Chang S-F. Using relevance feedback in content-based image Metasearch. IEEE Internet Comput 1998; 2(4):59–69.

[12]. Lu Tzu-Chuen, Chang Chin-Chen. Color image retrieval technique based on color features and image bitmap. Inf Process Manage 2007; 43:461–72.

[13]. H. Yu, M. Li, H.-J. Zhang, J. Feng, Color texture moments for content-based image retrieval, in Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 3, IEEE, 2002, pp. 929–932.

[14]. S. Liapis, G. Tziritas, Color and texture image retrieval using chromaticity histograms and wavelet frames, Multimedia, IEEE Transactions on 6 (5) (2004) 676–686.

[15]. G.-H. Liu, J.-Y. Yang, Content-based image retrieval using color difference histogram, Pattern Recognition 46 (1) (2013) 188–198.

[16]. P. Hiremath, J. Pujari, Content based image retrieval using color, texture and shape features, in Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on, IEEE, 2007, pp. 780–784.

[17]. C. Bai, W. Zou, K. Kpalma, J. Ronsin, Efficient color texture image retrieval by a combination of color and texture features in the wavelet domain, Electronics letters 48 (23) (2012) 1463–1465.

[18]. R. Kwitt, P. Meerwald, A. Uhl, Efficient texture image retrieval using copulas in a Bayesian framework, Image Processing, IEEE Transactions on 20 (7) (2011) 2063–2077.

[19]. A. D. El Maliani, M. El Hassouni, N.-E. Lamar, Y. Berthoumieu, D. Aboutajdine, Color texture classification using rao distance between multivariate copula based models, in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Computer Analysis of Images and Patterns, Springer, 2011, pp. 498–505.

[20]. T. S. Lee, Image representation using 2d Gabor wavelets, Pattern Analysis and Machine Intelligence, IEEE Transactions on 18 (10) (1996) 959–971.

[21]. F. Riaz, M. Areia, F. B. Silva, M. Dinis-Ribeiro, P. P. Nunes, M. Coimbra, Gabor textons for classification of Gastroenterology images, in Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on, IEEE, 2011, pp. 117–120.

[22]. L. Zhang, M. Fisher, W. Wang, Retinal vessel segmentation using Gabor filter and textons, MIUA 2014 (2014) 155–160.

[23]. C. Li, G. Duan, F. Zhong, Rotation invariant texture retrieval considering the scale dependence of Gabor wavelet, Image Processing, IEEE Transactions on 24 (8) (2015) 2344–2354.

[24]. Elasnaoui, K., Aksasse, B., and Ouanan, M., (2015), ”Color Image Retrieval Based on a Two-Dimensional Histogram”, International Journal of Mathematics and Computation, Vol. 26; Issue Num. 2, pp 10-18.

[25]. Elasnaoui, K., Aksasse, B., and Ouanan, M., (2015-a), ”Content-based color image retrieval based on the 2-D histogram and statistical moments”, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation.