



Digital Image Retrieval Using Combination of Texture, Shape and Colour Features

Mohammad Saberi¹, Farshad Tajeripour², and Shervan Fekri-Ershad³

^{1,2,3} Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

ABSTRACT

A robust method is proposed in this paper to retrieve digital images using combination of color, texture and shape features. The proposed method consists of three stages. Shape detection is done based on Top-Hat transform to detect and crop object part of the image in the first stage. The second is included a texture feature extraction algorithm using color local binary patterns (CLBP) and local variance features together. Finally, to retrieve mostly closing matching images to the query, log likelihood ratio is used as non-similarity ratio. The performance of the proposed approach is evaluated using Corel and Simplicity image sets and it compared by some of other well-known approaches in terms of precision and recall which shows the superiority of the proposed approach. Shift invariant, Low noise sensitivity, rotation invariant are some of other advantages.

KEYWORDS

Digital Image retrieval, Local binary pattern, Local variance, Object Cropping, Top-Hat Transform, Texture analysis

1. INTRODUCTION

In order to find the relevant images from information databases created due to recent technological developments in diversity of domains is an important field of research which calls image retrieval. The need to find a desired digital image from a collection of digital databases is shared by many professional groups, including journalists, forensic experts, design engineers, forensic experts and art historiographers.

Digital image retrieval approaches may be divided in two basic groups which calls Content based image retrieval (CBIR) and Text based image retrieval (TBIR). CBIR is a technique to retrieve images on the basis of image specific features such as color, texture and shape. In CBIR, initially, features are computed for both stored and query images, and used to declare images most closely matching the query.

Color-based retrieval first evolved from simple statistical measures such as average color, color histograms and spatial color descriptors [1]. Unfortunately, color histograms have limited discriminative power with respect to the spatial organization of colors. The other popular color features in content-based image retrieval applications include color correlograms [2], color moments [3] and MPEG-7 color descriptors [4]. The color correlogram describes the



probability of finding color pairs at fixed pixel distances and provide an efficient spatial retrieval feature

Partly due to the imprecise understanding and definition of what exactly visual texture actually is, texture features have an even larger variety than color features. As for the texture feature extractors, many approaches have been proposed based on Gabor filtering [5] Micro-Structure [6], GLCM [7], and Primitive texture features [8].

These texture features try to capture the characteristics of the image or image parts with respect to changes in certain directions and the scale of the changes. This is most useful for regions or images with homogeneous texture. Again, invariances with respect to rotations of the image, shifts or scale changes can be included into the feature space but information on the texture can get lost in this process.

Shape is a key attribute of segmented image regions, and its efficient and robust representation plays an important role in retrieval. Synonymous with shape representation is the way in which such representations are matched with each other. Some researchers are used shape features to retrieve similar images. In this respect, many techniques are used for shape detection such as morphological operations [9], edge detection [10], primitive shape filters [11] and image segmentation [12].

In this paper, we proposed an approach using combination of color, texture and shape features together. It is included two main steps. First one is prepared to crop the main object part of the query and database images based on top-hat transform as a shape detector. First step eliminates irrelevant parts of the images to extract more valid feature vector. Second step is included an algorithm for extracting features from object part image. Color local binary patterns (CLBP) and local variance are used as feature extractors in this step.

According to the size of huge image databases, we proposed a novel extra algorithm to reduce computational complexity. It is done as a preprocess step before retrieving to disregard the absolutely not close categories to the query. The proposed preprocess algorithm compares non-similarity amount between query image and each category to disregard nowhere near categories.

The performance of the proposed approach is evaluated by applying on Corel and Simplicity image sets in terms of precision and recall and it is compared by some other state of the art CBIR algorithms. Low noise sensitivity, rotation invariant, gray scale invariant, shift invariant and Low computational complexity are some other advantages of proposed approach which are demonstrated in the result section.

2. Proposed Object Cropping Algorithm:

2.1. Top-Hat Transform

Object cropping is one of the new research fields in image processing and computer vision. The aim of object cropping is to detect and cut the object parts of a complex image. Object parts of an image are those parts where meaningful information is proposed. For example in Fig.1, an image is shown which contains red flowers and leafs, but the meaningful information is in flower parts, so the object cropping algorithm must crop these parts form leafs. According to the image sets which are used in image retrieval problems, there are just one or some more objects in the images and all of other regions are not relevant. The irrelevant regions may invalid the values in extracted feature vectors. An approach to extract and crop object parts is



so necessary. Shape detection techniques detect objects in the images. In this paper an algorithms is proposed to detect and crop object part in input images based on one of popular shape detection techniques which calls top-hat transform.



Figure1. An examples of complex image

The groups of image processing operations which process the image based on shapes are referred as Morphological operations. In morphological operations the output image is created with help of applying structuring element to the input image. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors [20].

Dilation and erosion are the most basic morphological operations. Dilation creates the swelling effect of shapes of the objects by adding pixels to the boundaries of objects in an image, while erosion forms the object shape shrinking effect by removing pixels on object boundaries [20]. The size and shape of structuring element decide number of pixels added or removed from the objects in an image.

Dilation of a grey-scale image $I(x, y)$ by a grey-scale structuring element $B(s, t)$ is denoted by:

$$I \oplus B(x, y) = \text{Max} \{I(x - s, y - t) + b(s, t)\} \quad (1)$$

The domain of $I \oplus B$ is the dilation of the domain of "I" by the domain of "B". Erosion of a grey-scale image $I(x, y)$ by a grey-scale structuring element $B(s, t)$ is denoted by:

$$I \ominus B(x, y) = \text{min} \{I(x + s, y + t) - b(s, t)\} \quad (2)$$

The domain of $I \ominus B$ is the erosion of the domain of "I" by the domain of "B".

Opening operator is erosion followed by dilation and closing operator is dilation followed by erosion. Opening generally smoothes the contour of an image, breaks narrow gaps. As opposed to opening, closing tends to fuse narrow breaks, eliminates small holes, and fills gaps in the contours.

In mathematical morphology and digital image processing, top-hat transform is an operation that extracts small elements and details from given images. There exist two types of top-hat transform: The *white top-hat transform* is defined as the difference between the input image and its opening by some structuring element (Eq. 3). The *black top-hat transform* is defined dually as the difference between the closing and the input image (Eq. 4).



The white top-hat transform of input image "I" is given by:

$$T_1 = I - (I \circ S) \quad (3)$$

The Black top-hat transform of input image "I" is given by:

$$T_2 = (I \ S) - I \quad (4)$$

Where "I" means the input image and "S" is the structure element. "T₁" shows the white top-hat transform output and "T₂" shows the black top-hat transform output. Also, "○" denotes the opening operation and " " denotes closing operation.

2.2. Proposed Object Cropping Algorithm

In order to detect and crop the object parts in original input image, an edge detection technique should be used. In the previous section, black (white) top-hat transform is described as an edge detection technique. In order to crop object parts, an algorithm is proposed in this section which consists 3 sub-steps as follows:

- I) Applying black (white) top-hat transform on input image as edge detection technique to detect edge pixels of the input image. The output result is called "Edge-detected image".
- II) In order to crop the object parts of the original input image, it's enough to detect zone of the object region. The zone of the object region is limited between four bands (Up, down, left and right). The second step is detecting each of bands individually in edge-Detected image
- III) Cropping each pixel in the original image which is limited in the object zone using computed bands.

In the Fig.2, some results of applying the top-hat transform on Simplicity set images to detect and crop object parts are shown.

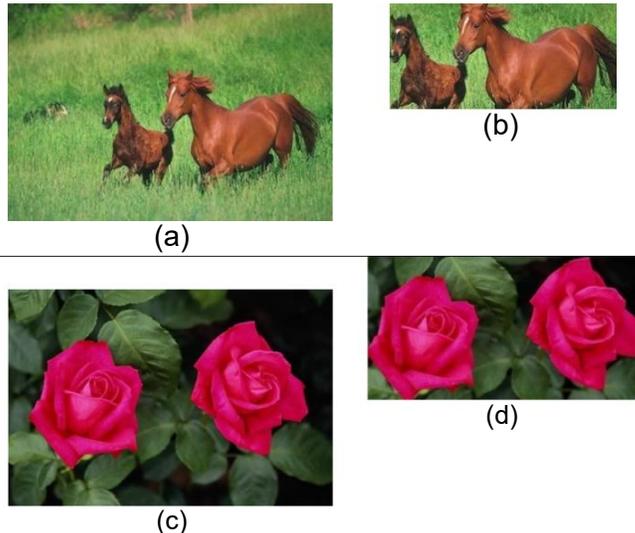


Figure2. Some examples of object cropping using Corel dataset images

(a) Original of Horses category (b) Object cropped of (a), (c) Original of Flower category (d) Object cropped of (c)



3. Texture Feature Extraction:

3.1. Local Binary Patterns

The local binary patterns (LBP) is a non-parametric operator which describes the local contrast and local spatial structure of an image. First time [13] introduced this operator and showed its high severability and discriminative power for texture analyzing and classification. Now, LBP is used as a good texture analysis method in various image processing aspects such as defect detection [17, 25], face recognition [26] and etc.

In order to evaluate the LBP, at a given pixel position (x_c, y_c) , LBP is defined as an ordered set of binary comparisons of pixel intensities between the central pixel and its surrounding pixels. Usually to achieve the rotation invariant, neighbourhoods would be assumed circular. Points which the coordination's are not exactly located at the center of pixel would be found by interpolation. Some of the circular neighbourhoods by different radius (R) and (P) number of neighbourhood's pixels are shown in Fig.3. Now, the LBP are defined at a neighbourhood of image by Eq.5.

$$\phi$$

Where, "g_c" corresponds to the grey value of the central pixel and "g_k" to the grey values of the neighbourhood pixels. P will be the number of neighbourhoods of central pixel, and function $\phi(x)$ is defined as:

$$\phi$$

!

Function $\phi(x)$ is labelled 1 each neighbour which has intensity value greater than central pixel and is labelled 0 to others. An example of applying LBP_{8,1} operator is shown in Fig.4. Where LBP_{8,1} amount of the center pixel is computed.

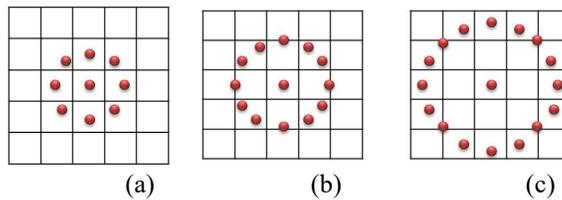


Figure3. Some examples of circular neighborhoods
 (a) P=8 R=1 (b) P=12 R=1.5 (c) P=16 R=2

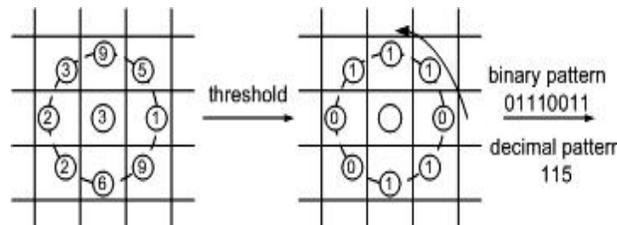


Figure4. An example of LBP_{8,1} texture model



3.1.1. Modified Local Binary Patterns

Practical experience of the authors in [14], however, has shown that LBPROT as such does not provide very good discrimination, and the computation complexity of basic LBP is high. To solve these problems, [15] defined a uniformity measure "U", which corresponds to the number of spatial transitions (bitwise 0/1 changes) in the "pattern". It is shown in Eq.7. For example, patterns 00000000 and 11111111 have U value of 0, while 11001001 have U value of 3.

$$U(\text{LBP}_{P,R}) = |\phi(g_1 - g_c) - \phi(g_p - g_c)| + \sum_{k=2}^p |\phi(g_k - g_c) - \phi(g_{k-1} - g_c)| \quad (7)$$

In this version of LBP, the patterns which have uniformity amount less than U_T are grouped as uniform patterns and the patterns with uniformity amount more than U_T grouped as non-uniform patterns. Finally, the LBP is computed by using Eq.8.

$$\begin{cases} \sum_{k=1}^p \phi(g_k - g_c) & \text{if } U(\text{LBP}_{P,R}) \leq U_T \\ P + 1 & \text{elsewhere} \end{cases} \quad (8)$$

Superscript " riu_T " reflects the use of rotation invariant "uniform" patterns that have U value of at most U_T .

According to Eq.8, applying $\text{LBP}_{P,R}$ will assign a label from "0" to "P" to uniform patterns and label "P+1" to non-uniform patterns. Because, $\text{LBP}_{P,R}$ just one label (P+1) is assigned to all of the non-uniform patterns, so uniform labels should cover mostly patterns in the image. In [16, 17] show that if in the definition of $\text{LBP}_{P,R}$ operator the value of U_T is selected equal to (P/4), only a negligible portion of the p patterns in the texture takes label P+1.

3.1.2. Proposed feature extraction based on LBP

As it was described in section 3.1.1, a label is assigned to each neighborhood. Regarding the Eq.9, if the number of neighbors is considered "P" pixels, applying $\text{LBP}_{P,R}$ will assign a label from zero to "P" to uniform segments and label "P+1" to non-uniform segments. So, for every image, one "P+2" dimensional feature vector can be extracted.

To extract the feature vector, first the $\text{LBP}_{P,R}$ is applied on the image and the labels are assigned to neighbors. Then the occurrence probability of each label in the image is regarded as one of the dimensions of the feature vector. The occurrence probability of a specific label in the image can be approximated by the ratio of the number of that label to the number of all labels (Eq.9).

$$P_k = \frac{N_{P_k}}{N_{\text{total}}} \quad 0 \leq k \leq P + 1 \quad (9)$$

Where, N_{P_k} is the number of neighbors that labeled as P_k , and N_{total} is the number of all neighbors. The extracted feature vector is shown in Eq.10.



$$\mathbf{F} \cdot \mathbf{F} = \langle P_0, P_1, \dots, P_{p+1} \rangle \quad (10)$$

Where, F is the feature vector extracted for neighbors and P_i is the occurrence probability of label "i" in neighbors.

3.2. Local Variance

The $LBP_{P,R}$ operator is a gray-scale invariant measure, i.e., its output is not affected by any monotonic transformation of the gray scale. It is an excellent measure of the spatial pattern, but it, by definition, discards contrast [15]. If gray-scale invariance is not required and it is wanted to incorporate the contrast of local image texture as well, we can measure it with a rotation invariant measure of local variance (Eq. 11):

$$\text{Where } \mu = \frac{1}{P} \sum_{k=1}^P g_k \quad (11)$$

$VAR_{P,R}$ is by definition invariant against shifts in gray scale. Since $LBP_{P,R}$ and $VAR_{P,R}$ are complementary, their joint distribution $LBP_{P,R} / VAR_{P,R}$ is expected to be a very powerful rotation invariant measure of local image texture.

3.2.1. Feature extraction based on $VAR_{P,R}$

A vector showing the variance of all points in the range is calculated by applying variance operator ($VAR_{P,R}$) on a neighbor in size of "P" and "R", for all points. After sorting this vector, it is divided into "K" bins and the number of variances falling in each bin is calculated. The final vector shows the number of similar variances in all "K" bins. This vector is represented as $VAR_{P,R}^K$.

The $VAR_{P,R}^K$ is a discriminative texture feature vector which includes the local contrast information of the input image. The results show the discrimination strength of this operator in retrieving images.

4. Proposed Image Retrieval approach

According to introduction part, content-based image retrieval which signifies by CBIR in shorthand form is a technique to retrieve images on the basis of image specific features such as color, texture and shape. In CBIR, initially, features are computed for both all of the stored and query images, and used to declare images most closely matching the query. The general block diagram of CBIR algorithms is shown in Fig.5.

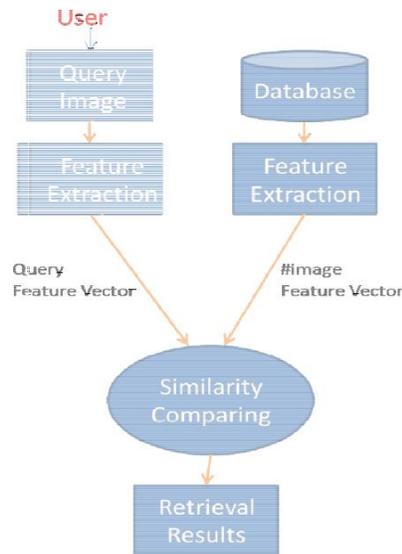


Figure 5. CBIR general Block Diagram

According to the Fig.5, the main step of the CBIR algorithms is feature extraction. In the proposed approach, the feature extraction step is included 3 sub-steps which is described ones by ones as follows:

I) Cropping the object parts of the query and database images using the algorithm which is proposed in the section 2.2. In this paper, the object cropped image is called OCI which is shorthand form of the Object-cropped Image.

II) The OCIs should be converted to the RGB color space (R=red, G=green and B=blue). Next, in order to extract texture features from the OCI, LBP_{P,R} should be applied on each color channel of the OCIs individually and extracting features based on the proposed approach in section 3.1.2 from each color channel.

After considering color features, to provide a main discriminated feature vector for each OCI, the Eq.(12) should be used which is described as follows:

$$3 \quad (12)$$

Where, F_{CLBP} is the output feature vector and F_d is shown the extracted feature vector for color channel "d". Also, "i" is the dimension number of the feature vectors.

III) Applying the VAR_{P,R} on OCIs and extracting features based on the proposed approach in section 3.2.1. The extracted feature vector is called F_{VAR} .

As it was mentioned in this section, the image analysis and feature extraction technique is done considering color, texture and shape features together. Using the proposed feature extraction algorithm, two feature vectors are provided for each input images either query or database images. The total block-diagram of the proposed feature extraction algorithm is shown in the Fig.6.

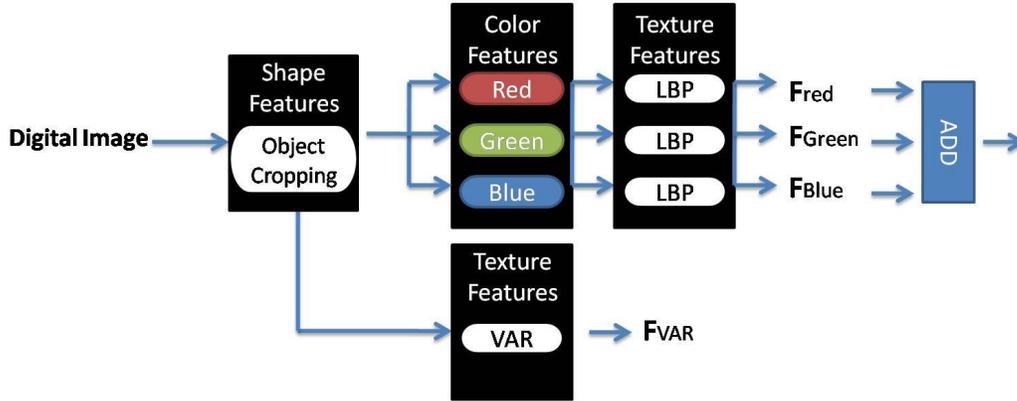


Figure6. The block- Diagram of the proposed feature extraction algorithm

According to the Fig.5, after feature extraction another important step is similarity comparing, to retrieve database images which are more close to the query.

The extracted feature vectors are probabilistic and sum of all dimension values are one. In [17], Tajeripour and Fekri-Ershad are used log-Likelihood to evaluate non-similarity amount between extracted local binary patterns feature vectors. In this respect, after researching about distance and similarity criteria, the log-likelihood ratio is selected as distance measure for similarity comparing step [15].

Log-likelihood ratio between two probabilistic feature vector (A, B) is computed as described in the Eq.(13).

$$L(A, B) = \sum_{k=1}^K A_k \log\left(\frac{A_k}{B_k}\right) \quad (13)$$

Where, L is the log-likelihood ratio and K is number of dimensions of the A and B vectors. Where A is shown the feature vector which extracted for the database image and B is shown the feature vector which extracted for query image. According to the proposed approach the summation of all dimensions value of A and B is equal one. It is shown as following:

$$\sum_{i=0}^{P+1} i = \sum_{i=0}^{P+1} K_i = 1 \quad (14)$$

L (A, B) is the log likelihood ratio between A and B. As it is seen in Eq(13), the log likelihood ratio is positive for each case because of being the positive value in each dimension of A and B vectors. The minimum amount of log likelihood ratio is zero which is shown the maximum similarity between database image and query feature vectors.

To declare database images most closely matching the query, the log-likelihood ratio is computed between query and each database image extracted feature vectors. Finally, the total similarity amount is considered sum of two computed ratios. It is shown in the Eq. (15).

$$L(Q, I) = L(F_{Q,CLBP}, F_{I,CLBP}) + L(F_{Q,VAR}, F_{I,VAR}) \quad (15)$$



Where, $L(Q, I)$ is shown the similarity amount between query image (Q) and database image (I). F_{CLBP} and F_{VAR} are two extracted feature vectors which are evaluated based on $LBP_{P,R}$ and $VAR_{P,R}$.

5. Results

The performance of the image retrieval approach was tested upon the following two popular databases: (a) Simplicity images and (b) COREL 5,000 miscellaneous database which are downloaded from [21].

The Corel image database contains a large amount of images of various contents ranging from animals and outdoor sports to natural scenarios. The Corel contains 50 categories. There are 5000 images from diverse contents such as fireworks, bark, microscopic, tile, food texture, tree, wave, pills and stained glass. Each category contains 100 images of size 192×128 in JPEG format.

The Simplicity database includes 10 different categories. These categories are titled: Beach, Africa, Elephants, Buildings, Buses, Dinosaurs, Horses, Flowers, Food and Mountains. Each category has absolutely 100 images, along with some images undergoing changes due to rotation, translation, scaling, noise injection, illumination, etc. The total number of simplicity database size is 1000.

In order to evaluate the effectiveness of the proposed approach, the Precision and Recall curves are adopted, which are the most common criteria used for evaluating image retrieval performance.

Precision and Recall are defined as follows:

$$\text{Precision (N)} = I_N / N \quad (16)$$

$$\text{Recall (N)} = I_N / M \quad (17)$$

Where " I_N " is the number of images retrieved in the top " N " positions that are similar to the query image, " M " is the total number of images in the database similar to the query (each category size), and " N " is the total number of images retrieved.

In this respect, in order to evaluate the performance, N is considered 10, 20, 30 and 40 for both of test datasets. Also, for feature extraction, different sizes was tested for CLBP and VAR, $P = 8$ and $K = 16$ provided maximum accuracy.

According to the Corel and Simplicity categories, " M " is 100. The average retrieval precision and recall for Corel dataset are shown in the Table 1.

In order to compare our approach, some of other previous well-know approaches [18] and [19] were survived.

In [18], an approach is proposed for CBIR using visually significant point features. The authors [18], were evaluated their approach performance using simplicity dataset considering $N = 10, 20$ and 40. The precision% is computed for each category separately. The average precision of their approach [18] is shown in the Table 2 and it was compared by our proposed approach.

As seen from Table2, our proposed approach has reported better results than the [18] method for 7 categories and total average. G. H. Liu et al are prepared an algorithm based on multi-texton histograms [19]. They evaluated their CBIR system performance using Corel dataset and



considering N=12. The average precision and recall based on [19], are shown in the Table 3. It is compared by proposed approach in a same situation (M= 5000, N=12). It can be seen from the Table 2 & 3, that our method achieves much better results than [18] and [19] methods.

Table1. The average precision and recall of Proposed approach on Corel (5000s) dataset

Dataset Retrieved Images	Precision	Recall
N = 10	53.62	5.36
N = 20	51.64	10.32
N = 30	50.24	15.07
N = 40	46.12	18.44

Table2. The average precision% of proposed and [18] on Simplicity

Approach Category	[18]			Proposed		
	N=10	N=20	N=40	N=10	N=20	N=40
Africa	70.50	61	58	67.34	61.04	56.76
Beach	71.48	56.23	52.58	73.10	59.11	53.87
Building	72.40	63.67	56	76.48	68.72	60.64
Bus	81.45	72.77	62.67	93.23	91.47	71.83
Dinosaur	100	95	92	100	98.87	95.48
Elephant	85	77	66	80	75.92	61.5
Flower	92	83	71.2	94.58	89.44	73.5
Horses	100	95	83	92.28	89.76	79.62
Mountains	72	68	62	72.4	68.38	63.24
Food	62	57	55.2	63.96	59.49	55.78
Average	80.68	72.86	65.86	81.33	76.22	67.22

Table3. The average precision% and Recall% of proposed and [19] on Corel, N=12

Measure Approach	Precision	Recall
[19]	49.98	6.00
Proposed	53.04	6.36

As seen from Table3, our proposed approach has reported better average precision% than the [19] and the recall averages are near same.



6. Conclusion

The specific goal of this paper was to prepare a robust approach for digital image retrieval. In this respect, an approach is proposed using combination of texture, shape and color features together unlike previous methods which are work using one of the feature extraction domains. The proposed approach is consists two main steps. In the first one, feature extraction is done using combination of texture, shape and color features. In the second step, log likelihood ratio is used to declare most closely matching images from a huge database to the query. The performance of the approach is evaluated using Corel and Simplicity image sets and it compared some of other well-known approaches in terms of precision and recall. The results are shown the high performance of the proposed approach to retrieve certainly same category images by the query. Some of other advantages of the proposed approach are as follows:

- I) the proposed digital image retrieval approach is rotation invariant which is due to using local binary pattern and local variance [15] in feature extraction step.
- II) The precision and recall amount of the proposed CBIR system is higher than many other well known CBIR algorithms.
- III) In this paper, a novel feature extraction is described using combination of top-hat transforms, local binary patterns and local variance that can be used for other image processing cases such as face recognition, defect detection and image parsing to analyze the digital color images.

References

- [1] Pass G., and Zabih R., (1999) "*Comparing Images Using Joint Histograms*", ACM Journal of Multimedia Systems, **7(3)**, pp. 234-240
- [2] Ojala T., Rautiainen M., Matinmikko E., and Aittola, M., (2001), "*Semantic Image Retrieval with HSV Correlograms*", In Proc. of 12th Scandinavian Conference on Image Analysis
- [3] Velmurugan K., and Santhosh Baboo S., (2011), "*Content-Based Image Retrieval using SURF and Color Moments*", Global Journal of Computer Science and Technology, **11(10)**, pp. 1-4
- [4] Eidenberger, H., (2004), "*statistical analysis of MPEG-7 image descriptions*", ACM Multimedia Systems journal, **10(2)**, pp.84-97
- [5] Manjunath B.S., and Ma W.Y., (1996) "*Texture features for browsing and retrieval of image data*", IEEE Transaction on pattern analysis and machine intelligence, **18**, pp. 837-842.
- [6] Guang-Hai L., Zuo-Yong L., Lei Z., and Yong X., (2011), "*Image retrieval based on micro-structure descriptor*", Pattern Recognition, **44(9)**, pp. 2123-2133
- [7] Jhanwar N., Chaudhuri S., and Zavidovique B., (2004), "*Content based image retrieval using motif co-occurrence matrix*", Image and Vision Computing, Vol. 22, No. 14, pp.1211-1220
- [8] Hsin-Chih L., Ling-Ling W., and Shi-Nine Y., (1999), "*Regular-texture image retrieval based on texture-primitive extraction*", Image and Vision Computing, **17(1)** , pp.51-63
- [9] Kekre H.B., Mukherjee P., and Wadhwa SH., (2010), "*Image Retrieval with Shape Features Extracted using Morphological Operators with BTC*", International Journal of Computer Applications, **12(3)**, pp. 1-5
- [10] Berman A.P., and Shapiro, L.G., (1999), "*A flexible image database system for content-based retrieval*", Computer Vision and Image Understanding, **75**, pp. 175-195



- [11] Greenspan S., Hayit, Malik, Jitendra, and Puzicha, (2002), "*Shape matching and object recognition using shape contexts*", IEEE Transactions on Pattern Analysis and Machine Intelligence, **24(4)**, pp. 509-522
- [12] Lifeng L., and Sclaroff S., (1999), "*Automatic Deformable Shape Segmentation for Image Database Search Applications*", In Proc. of International Conference on Visual Information
- [13] Ojala, T., Maenpaae, T., and Pietikainen, M., (2000), "*Texture Classification by Multi Predicate Local Binary Pattern Operators*", In Proc. 15th International Conference on Pattern Recognition, **3**, pp. 951-954.
- [14] Pietikäinen, M., Ojala, T., and Xu, Z., (2000), "*Rotation-Invariant Texture Classification Using Feature Distributions*", Pattern Recognition, **33**, pp 43-52.
- [15] Ojala, T., Pietikainen, M., and Maenpaa, T., (2002), "*Multi resolution gray-scale and rotation invariant texture classification with local binary patterns*", IEEE Transactions on Pattern Analysis and Machine Intelligence, **24(7)**, pp. 971–987.
- [16] Tajeripour, F., Kabir, E., and Sheikhi, A., (2008), "*Fabric Defect detection using Modified Local Binary Patterns*", EURASIP Journal on Advances in Signal Processing, **1**, pp. 1-12.
- [17] Fekri-Ershad, Sh., and Tajeripour, F., (2012), "*A Robust Approach for Surface Defect Detection Based on One Dimensional Local Binary Patterns*", Indian Journal of Science and Technology, **5(8)**, pp.3197-3203
- [18] Banerjee, M., Malay, K.K., Maji, P., (2009), "*Content-based image retrieval using visually significant point features*", Fuzzy Sets and Systems, **160**, pp. 3323-3341
- [19] Guang-Hai, L., Lei, Z., Ying-Kun H., Zuo-Yong, Li., Jing-Yu, Y., (2010), "Image retrieval based on multi-texton histogram", Pattern Recognition, **43**, pp. 2380-2389
- [20] Chen, T., Wu, Q.H., Rahmani-Torkaman, R., Hughes, J., (2002), "A pseudo top-hat mathematical morphological approach to edge detection in dark regions", Pattern Recognition, **35**, pp. 199-210
- [21] <http://bergman.stanford.edu/cgi-bin>
- [22] Randen, T., and Husoy, J.H., (1999), "*Filtering for Texture Classification: A Comparative Study*", IEEE Trans. Pattern Analysis and Machine Intelligence, **21(4)**, pp. 291-310
- [23] Wu, J.K. and Narasimhalu A.D., (1998) "*Fuzzy content based retrieval in Image database*", Information Processing and Management, **34(5)**, pp. 513-534,
- [24] Santini, S., Jain R., (1999), "*Similarity Measure*", IEEE Transaction on Pattern Analysis and Machine Intelligence, **Sep**, pp. 871-883
- [25] Tajeripour, F., and Fekri-Ershad, Sh., (2012), "*Porosity Detection by using improved Local Binary Pattern*", In Proc. of the 11th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA '12), , pp. 116-121
- [26] Liao, S., Zhu, X., Lei, Z., Zhang, L., and Li, S.Z., (2007), "*Learning Multi-scale Block Local Binary Patterns for Face Recognition*", In Proc. of IAPR/IEEE International Conference on Biometrics, pp. 828-837