



# Tetragonal Local Octa-Pattern (T-LOP) based image retrieval using genetically optimized support vector machines

Zain Shabbir<sup>1</sup>  · Aun Irtaza<sup>2</sup> · Ali Javed<sup>3</sup> · Muhammad Tariq Mahmood<sup>4</sup>

Received: 25 February 2018 / Revised: 5 January 2019 / Accepted: 3 April 2019 /

Published online: 7 May 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

The enormous increase in digital image collections motivates the research community to propose powerful Content Based Image Retrieval (CBIR) algorithms to employ in critical scientific domains. In this paper, we have proposed Tetragonal Local Octa-Patterns for CBIR that are based on the direction of center pixel and generate an 8-bit octa-pattern. Neighbors at three diagonal locations are then used to generate Tetragonal Octa-Patterns. In order to enhance the precision, Genetic algorithm has been applied on obtained features to resolve the class imbalance problem for better classification through SVM. Experimental results prove the reliability of method by comparing against state-of-the-art methods in terms of precision and recall.

**Keywords** Tetragonal Local Octa-Pattern · Image retrieval · Genetic algorithm · SVM

---

✉ Zain Shabbir  
zain.shabbir@uet.edu.pk

Aun Irtaza  
aun.irtaza@uettaxila.edu.pk

Ali Javed  
ali.javed@uettaxila.edu.pk

Muhammad Tariq Mahmood  
tariq@koreatech.ac.kr

<sup>1</sup> Department of Electrical, Electronics and Communication Engineering, University of Engineering and Technology Lahore (Faisalabad Campus), Faisalabad, Pakistan

<sup>2</sup> Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan

<sup>3</sup> Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan

<sup>4</sup> School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, South Korea

## 1 Introduction

Due to the extensive usage of digital cameras and availability of large image repositories, the association of textual annotation with images is a very difficult task. Hence, the retrieval of relevant information in such unorganized collections becomes almost impossible. Therefore, there is an extreme need of the image retrieval methods that can retrieve relevant images without relying on the text-based labeling pre-associated with the images. To address this issue, one such potential area of research is content-based image retrieval (CBIR) [36].

In CBIR, image retrieval occurs by exploring the visual information available in the images i.e. colors [6, 32, 34], shapes [13, 19, 30], texture [9, 25, 26] or the spatial layout [2, 24]. The main working theme of CBIR systems involves the representation of images in the form of feature vectors through available visual clues. These feature vectors are then searched against feature vector of query image either by computing the distance based similarity or training a semantic classifier. Although CBIR has found many applications in fields like surveillance, medicine, internet image and video search; but still the research is competing for more powerful solutions of CBIR.

The foremost reason behind the challenging nature of CBIR is that the semantics behind the query image are relative and vary with respect to users. As described in Fig. 1, both the output images against query image (Fig. 1a) are relevant depending on what user wants to search. Apart from this, even the various forms of relevant outputs are still unachievable if the image representation occurs poorly. Hence, to address these issues several research contributions have been presented specifically for image representation generation to ensure meaningful image retrieval.

In [18], texture analysis of images was performed through wavelet packets tree, and Eigen values obtained through the Gabor filter. The feature representation scheme didn't introduce the feature normalization to resolve large negative and positive values, which limits the image retrieval performance. In [17], the same feature representation scheme was enhanced by further analyzing the texture through Curvelet transform. However, the feature representation still suffers from the same limitations as that of the parent scheme. In [39], local derivative patterns (LDPs) were proposed by extending the LBP [29] representations for face recognition. LDPs compute the  $n^{\text{th}}$ -order local derivative patterns and generate the feature representations; however, LDPs were unable to adequately deal with the range of appearance variations attributed to the unconstrained natural images. To overcome this problem, local ternary patterns (LTPs) were proposed in [35] for face recognition under different lighting conditions.

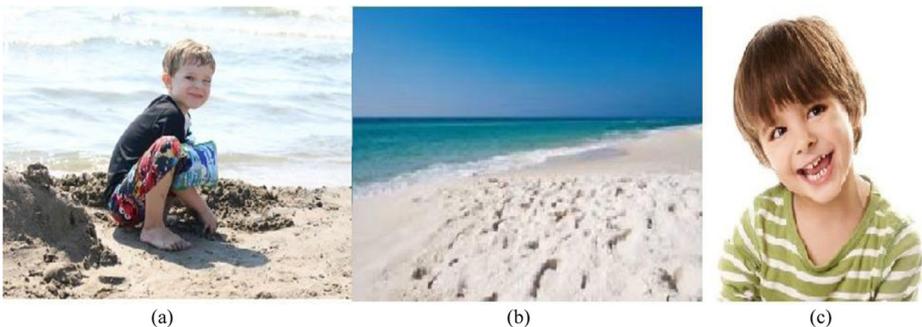


Fig. 1 Query image and retrieval results

However, the LTPs also encode the information only in horizontal and vertical directions for edge representation as previously done in LBPs and LDPs. In [28], local tetra patterns (LTrPs) extended the LTP idea by computing the relationships between referenced pixel and the neighborhood pixels based on the four edge directions. The LTrP image retrieval performance is better as compared to that of the existing methods.

The LBP, LDP, LTP, and LTrP encode the information in the form of features that are based on the distribution of edges. Exploring the images by differentiating edges in more directions is evidence for the improvement in the form of image retrieval [28]. Therefore, the emphasis of the paper is on the presentation of a feature representation scheme that is more elaborative and descriptive as compared to the existing methods and ensures the precise image retrieval output.

The elaborative image representation scheme we introduced in the form of Tetragonal Local Octa Patterns (T-LOP) differentiates the images in eight directions and contains the uniform patterns of various diagonals. Uniform patterns allow us to generate image representations that are rotation invariant. Whereas, the diagonals of various sizes make the representation scale invariant. The feature representation scheme thus has the attributes like rotation and scale invariance and provides a global representation of the Local Texture Patterns that make them more descriptive.

To further improve the image retrieval performance, the semantic class imbalance problem [10] has also been resolved in this paper through genetic algorithms (GA).

As described in [10], datasets with skewed class distribution usually tend to suffer from class overlapping that makes the classifier learning difficult. Furthermore, the evaluation criterion (which also guides the learning procedure) leads to ignore minority class examples, and thus, classification ability of induced classifier is lost. As an example, let us assume a bi-class dataset which has imbalance ratio 1:100; if a classifier tries to maximize the precision of classification rule, may attain precision of 90% by discounting only positive examples. If there are large number of highly correlated semantic classes as in CBIR, the severity of the problem increases greatly. If a semantic class has insufficient number of samples, the chances of incorrect association for query images increase as well. Thus, in this paper, we argue that the precise image retrieval cannot occur until the overlapping semantic class imbalance problem remains unresolved.

The rest of the paper is organized as follows: Section 2 briefly describes the comparative techniques used in this work. Section 3 deals with Local Patterns that provide the basis of this work. Section 4 provides the details of the proposed algorithm. In Section 5, experimental results are provided. Finally the paper is concluded in Section 6.

## 2 Related work

Feature extraction is first step of any CBIR algorithm. These features are extracted using the visual contents of image e.g. texture, color, shape etc. Many researchers are working to develop a precise CBIR algorithm. In this regard, in [12] a Halftoning-Based Block Truncation Coding based algorithm for feature extraction was presented. While encoding, image is compressed into corresponding quantizers and bitmap image that are then used to derive image features. In [8], a new matching strategy was proposed for image retrieval. The strategy was based upon the minimum area between two vectors, which was then used to compute similarity value between query and test images of a class. In [17], hybrid texture features were proposed using Wavelet packets, Curvelet transform and Gabor filters. These features were

then combined to make hybrid feature vectors. A similar type of hybrid technique was used in [38] where Curvelet transform was integrated with region based vector codebook sub-band clustering for color extraction and texture analysis.

In [7], a feature selection technique that extracts the most relevant features was proposed that simplifies the computation and increases the retrieval rate. Authors in [23] again used hybrid technique for feature extraction. Color co-occurrence matrix and color histogram for k-mean were used; and a texture based technique that computes the difference between pixels of scan pattern was then combined for feature extraction. In [20], Motif Co-occurrence Matrix (MCM) was used for feature extraction. Although the technique is similar to Color Co-occurrence Matrix but as MCM captures third order image statistics in local neighborhood, therefore, authors believe that the technique improves the retrieval rate. In [16] composite sub-band gradient vector and energy distribution pattern string were used as feature vectors. A fuzzy matching approach was then used to remove undesired images against the query image. In [5], authors improved the work in Kolmogorov complexity-based similarity measures for texture matching problems. Campana-Keogh (CK) video compression based method was then proposed to measure the texture. Through these video compressors, Kolmogorov complexity was approximated and a new texture similarity measure, CK-1 was introduced. In [11], an image compression approach based on sparse representation was proposed to encode information in an image. The degree of compression was measured using the sparsity of representation that signifies the similarity between images. In [33], SIFT and Invariant Feature From Segmentation (IFFS) were used as image features and then structural representation in image retrieval was computed. In [37], authors proposed an image retrieval system through Adaptive Region Matching (ARM) that outperforms many state of the art methods.

### 3 Local patterns

#### 3.1 Local binary pattern (LBP)

LBP operator [29] is extensively utilized in micro-patterns for modeling the texture structure of images. The LBP extracts the discriminative features by encoding the binomial concatenation of the relationship between the intensities of a referenced pixel  $g_c$  and its surrounding neighborhood  $g_p$  with  $P$  neighbors (Fig. 2) as described in Eq. 1.

$$LBP = \sum_{p=1}^P 2^{(p-1)*} f_1(g_p - g_c) \quad (1)$$

Where  $f_1(x)$  can be computed as follows:

$$f_1(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{else} \end{cases} \quad (2)$$

In circular LBP, neighbors at different radius locations are considered for pattern calculation. The idea has been illustrated in Fig. 3.

In Fig. 3a, eight neighbors at radius = 1 while in Fig. 3b and c, 16 and eight neighbors at radius = 2 respectively have been considered for pattern calculation.

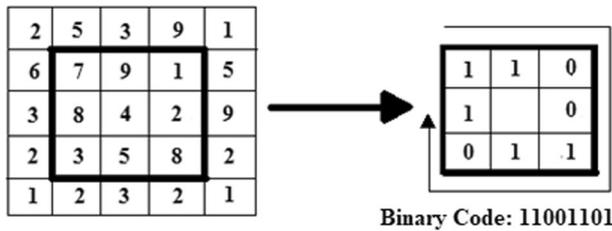


Fig. 2 Local binary pattern calculation

### 3.2 Local ternary pattern (LTP)

Local Ternary Patterns [35] (three value coding patterns) are further extension of LBP. In LTPs, gray values of pixels in a fixed zone  $\pm t$  around center pixel are quantized to zero. Pixels above the zone are quantized to +1 while below the zone are quantized to -1. So the function  $f_l(x)$  in LBP now takes the form:

$$f_2(x, g_c, t) = \begin{cases} +1, & x \geq g_c + t \\ 0, & |x - g_c| < t \\ -1, & x \leq g_c - t \end{cases} \tag{3}$$

Where  $x$  represents  $g_p$ , pattern thus achieved is divided into upper and lower ternary patterns. More detailed description can be found in [35].

### 3.3 Local derivative pattern (LDP)

The LDP [39] considers the LBP as a non-directional first order derivative pattern and performs its directional expansion through higher order derivatives to obtain more discriminative features. In order to obtain the  $n^{th}$  order LDP, the  $(n-1)^{th}$  order derivatives (denoted as:  $I_{\theta}^{n-1}(g_c)$ ) referring Fig. 4 are computed along  $0^\circ, 45^\circ, 90^\circ$  and  $135^\circ$  directions ( $\theta$ ) based on:

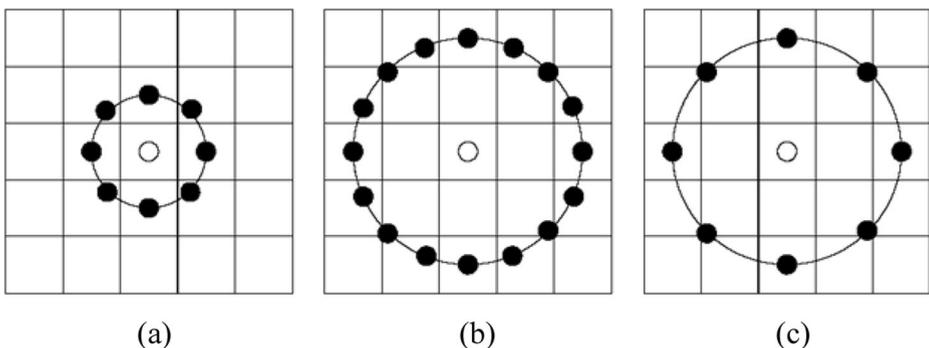
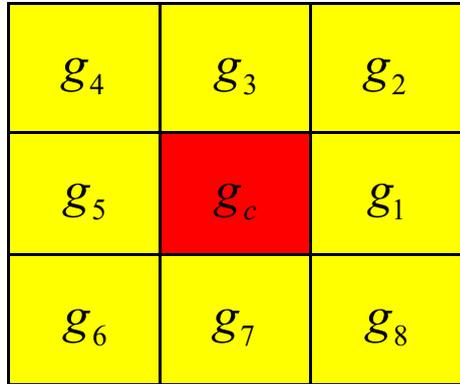


Fig. 3 Circular LBPs

**Fig. 4** Center pixel and its neighbors in image block



$$\begin{aligned}
 I_{0^0}^{n-1}(g_c) &= I_{0^0}^{n-2}(g_1) - I_{0^0}^{n-2}(g_c) \\
 I_{45^0}^{n-1}(g_c) &= I_{45^0}^{n-2}(g_2) - I_{45^0}^{n-2}(g_c) \\
 I_{90^0}^{n-1}(g_c) &= I_{90^0}^{n-2}(g_3) - I_{90^0}^{n-2}(g_c) \\
 I_{135^0}^{n-1}(g_c) &= I_{135^0}^{n-2}(g_4) - I_{135^0}^{n-2}(g_c)
 \end{aligned}
 \tag{4}$$

The  $n^{th}$  order LDP in  $\theta$  derivative direction for  $g_c$  is defined as:

$$LDP_{P,\theta}^n(g_c) = \left\{ f_3 \left( I_{\theta}^{n-1}(g_p), I_{\theta}^{n-1}(g_c) \right) \mid p = 1, 2, \dots, P \right\}
 \tag{5}$$

where

$$f_3 \left( I_{\theta}^{n-1}(g_p), I_{\theta}^{n-1}(g_c) \right) = \begin{cases} 0, & \text{if } I_{\theta}^{n-1}(g_p), I_{\theta}^{n-1}(g_c) > 0 \\ 1, & \text{if } I_{\theta}^{n-1}(g_p), I_{\theta}^{n-1}(g_c) \leq 0 \end{cases}
 \tag{6}$$

The output of the function  $f_3(\dots)$  represents the turning points labeled as 1 and monotonically increasing or decreasing values labeled as 0. Finally the  $n^{th}$  order LDP can be computed by the concatenation of all four 8 bit directional LDPs e.g. 2nd order LDP can be computed using Eq. given below.

$$LDP^2(g_c) = \{ LDP_{\theta}^2(g_c); \theta = 0^0, 45^0, 90^0, 135^0 \}
 \tag{7}$$

### 3.4 Local tetra pattern (LTrP)

LDP represents the 1D spatial relationship in the intensity window with single high-order derivate direction. Whereas, the main idea behind the LTrP [28] is that: more discriminative features can be extracted by extending the LDP in vertical and horizontal higher order derivate directions. After considering these observations we can infer that to extract more detailed information than that in LDP, one higher order derivative direction may be expanded to four distinct values. The  $n^{th}$  order LTrP pre-computes the  $(n - 1)^{th}$  order derivatives along  $0^{\circ}$  and  $90^{\circ}$  using Eqs. (8) and (9) to represent four distinct values at  $g_c$  i.e.:

$$I_{0^{\circ}}^{n-1}(g_c) = I_{0^{\circ}}^{n-2}(g_1) - I_{0^{\circ}}^{n-2}(g_c) \tag{8}$$

$$I_{90^{\circ}}^{n-1}(g_c) = I_{90^{\circ}}^{n-2}(g_3) - I_{90^{\circ}}^{n-2}(g_c) \tag{9}$$

$$I_D^{n-1}(g_c) = \begin{cases} 1, & I_{0^{\circ}}^{n-1}(g_c) \geq 0 \wedge I_{90^{\circ}}^{n-1}(g_c) \geq 0 \\ 2, & I_{0^{\circ}}^{n-1}(g_c) < 0 \wedge I_{90^{\circ}}^{n-1}(g_c) \geq 0 \\ 3, & I_{0^{\circ}}^{n-1}(g_c) < 0 \wedge I_{90^{\circ}}^{n-1}(g_c) < 0 \\ 4, & I_{0^{\circ}}^{n-1}(g_c) \geq 0 \wedge I_{90^{\circ}}^{n-1}(g_c) < 0 \end{cases} \tag{10}$$

Once the direction of the referenced pixel is computed, the  $n^{th}$  order LTrP can be defined as:

$$LTrP_p^n(g_c) = \{f_4(I_D^{n-1}(g_p), I_D^{n-1}(g_c))\} | p = 1, 2, \dots, P \tag{11}$$

Where

$$f_4(I_D^{n-1}(g_p), I_D^{n-1}(g_c)) = \begin{cases} 0, & \text{if } I_D^{n-1}(g_p) = I_D^{n-1}(g_c) \\ I_D^{n-1}(g_p), & \text{else} \end{cases} \tag{12}$$

Once the direction of a pixel is computed, the LTrP can further be segregated to 3 binary patterns. Let the direction of center pixel is 1 then LTrP can further be extended as follows:

$$LTrP_p^n(g_c)_{direction=\Delta} = \sum_{n=1}^N 2^{(n-1)*} f_5(LTrP_p^n(g_c))_{direction=\Delta} \tag{13}$$

Where in this particular case  $\Delta = \{2, 3, 4\}$  and,

$$f_5(LTrP_p^n(g_c))_{direction=\Delta} = \begin{cases} 1, & \text{if } LTrP_p^n(g_c) = \Delta \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

For each direction  $\Delta$ , an eight bit pattern is achieved. Hence, total of 3 eight bit patterns are achieved. Similarly, patterns can be achieved for other center pixel directions. A fourth binary pattern is calculated using the magnitude of horizontal and vertical derivatives of 8 neighboring pixels, known as magnitude pattern (MP), and is expressed as follows:

$$MP = \sum_{p=1}^P 2^{p-1*} f_6(M_{I(g_p)} - M_{I(g_c)}) \Big|_{P=8} \tag{15}$$

Where

$$M_{I(g_p)} = \sqrt{(I_{0^{\circ}}^{n-1}(g_p))^2 + (I_{90^{\circ}}^{n-1}(g_p))^2} \tag{16}$$

$$f_6(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{else} \end{cases} \tag{17}$$

These four binary patterns are used to describe the texture of image. More details about LTrPs feature representation can be found in [28].

### 4 Proposed approach

In our proposed algorithm, the texture representation capabilities of LTrPs are further extended by introducing the diagonal derivatives in the texture representation step, thus forming the *Local Octa Patterns (LOcPs)*. Moreover, neighbors around center pixel are considered at different diagonal locations for feature extraction (i.e. neighbors at diagonal: 3, 5 and 7), thus forming *Tetragonal Octa Patterns*.

Apart from the powerful image representation, which is a key requirement for semantically correct image retrieval output, the image retrieval performance of CBIR systems still degrades. The ultimate reason for the performance deficit is the availability of large number of semantic concepts and fewer training samples in the class of interest. Therefore, before training the semantic classifier (SVM) we have resolved this issue by applying the genetic algorithm (GA) on extracted features to reduce the semantic class imbalance problem. Thus, overall image retrieval framework as described in Fig. 5 ensures the improved image retrieval output against the query images.

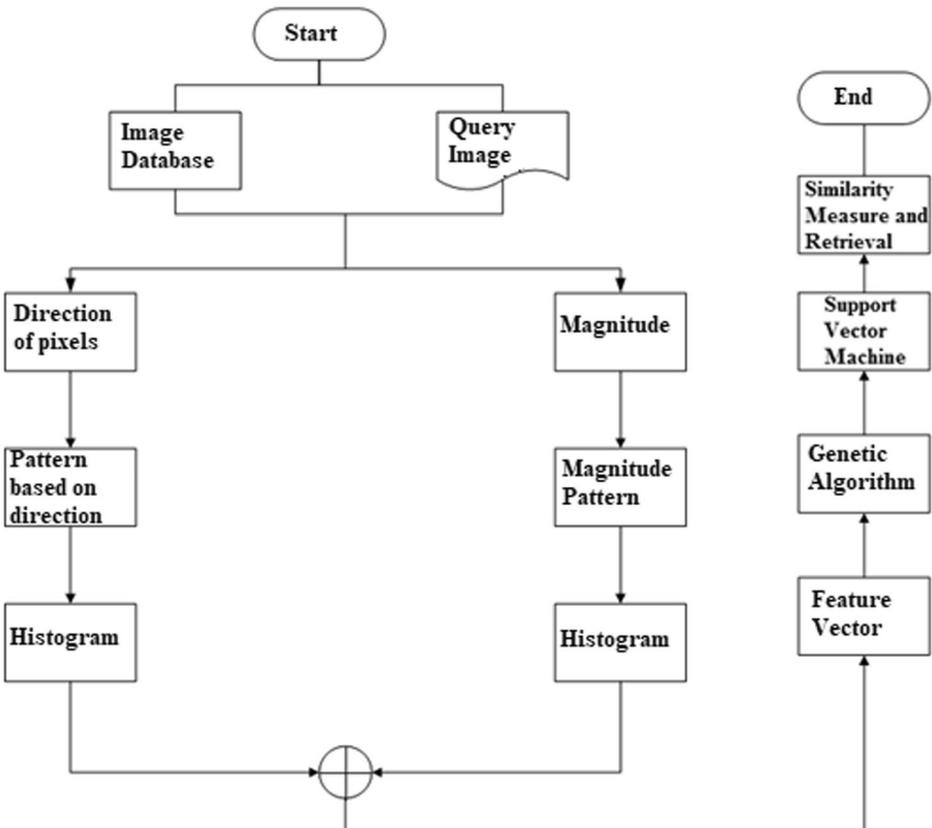


Fig. 5 Block diagram of proposed approach

### 4.1 Tetragonal Local Octa Pattern (T-LOP)

T-LOP descriptor considers the horizontal, vertical, and diagonal derivative directions and contains the uniform patterns of various diagonals to represent a region. Uniform patterns at one end reduce the feature vector length and also allow us to generate image representations that are rotation invariant. Whereas, the diagonals of various sizes make the representation scale invariant. The main motivation behind the utilization of the diagonal derivative direction along with LTRPs is to explore more details in images e.g. as LTRPs take into account only horizontal and vertical directions, therefore, when applied over the fundus images, depicted insensitivity against diagonally shaped blood vessels; hence did not encode them effectively. Therefore, to overcome this shortcoming we have extended the LTRPs by computing the four derivative directions instead of two directions.

The  $n^{th}$  order T-LOP descriptor pre-computes the  $(n - 1)^{th}$  order derivatives along  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  using Eq. (4) to represent eight distinct values at  $g_c$  as:

$$I_D^{n-1}(g_c) = \left\{ \begin{array}{l} 1, I_{0^\circ}^{n-1}(g_c) \geq 0 \wedge I_{90^\circ}^{n-1}(g_c) \geq 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) \geq 0 \\ 2, I_{0^\circ}^{n-1}(g_c) < 0 \wedge I_{90^\circ}^{n-1}(g_c) \geq 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) \geq 0 \\ 3, I_{0^\circ}^{n-1}(g_c) < 0 \wedge I_{90^\circ}^{n-1}(g_c) < 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) \geq 0 \\ 4, I_{0^\circ}^{n-1}(g_c) \geq 0 \wedge I_{90^\circ}^{n-1}(g_c) < 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) \geq 0 \\ 5, I_{0^\circ}^{n-1}(g_c) \geq 0 \wedge I_{90^\circ}^{n-1}(g_c) \geq 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) < 0 \\ 6, I_{0^\circ}^{n-1}(g_c) < 0 \wedge I_{90^\circ}^{n-1}(g_c) \geq 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) < 0 \\ 7, I_{0^\circ}^{n-1}(g_c) < 0 \wedge I_{90^\circ}^{n-1}(g_c) < 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) < 0 \\ 8, I_{0^\circ}^{n-1}(g_c) \geq 0 \wedge I_{90^\circ}^{n-1}(g_c) < 0 \wedge (I_{45^\circ}^{n-1}(g_c) \vee I_{135^\circ}^{n-1}(g_c)) < 0 \end{array} \right\} \quad (18)$$

Once the direction of the referenced pixel is computed, the  $n^{th}$  order T-LOPs can be defined as:

$$TLOP_P^n(g_c) = \left\{ f_7 \left( I_D^{n-1}(g_p), I_D^{n-1}(g_c) \right) \right\} | p = 1, 2, \dots, P \quad (19)$$

Where

$$f_7 \left( I_D^{n-1}(g_p), I_D^{n-1}(g_c) \right) = \begin{cases} I_D^{n-1}(g_p), & \text{if } I_D^{n-1}(g_p) \neq I_{Dir}^{n-1}(g_c) \\ 0, & \text{else} \end{cases} \quad (20)$$

From Eq. (19), we get T-LOP code that is further divided into 7 binary patterns based on the direction of center pixel.

$$TLOP_P^n | \left\{ \overline{D} | \forall D, \exists -I_D^{n-1}(g_c) \right\} = f_8(TLOP_P^n(g_c)) | \left\{ \overline{D} | \forall D, \exists -I_D^{n-1}(g_c) \right\} \quad (21)$$

$$f_8(TLOP_P^n(g_c)) | \overrightarrow{D \in \overline{D}} = \begin{cases} 1, & \text{if } TLOP_P^n(g_c) = \overrightarrow{D} \\ 0, & \text{else} \end{cases} \quad (22)$$

Where  $\bar{D}$  is the set of all quadrants except the quadrant of the referenced pixel and  $\vec{D}$  is one of the quadrant of  $\bar{D}$ . Afterwards, the T-LOP code can be generated as:

$$TLOP_p^n \left| \left\{ \bar{D} \mid \forall D, \exists -I_D^{n-1}(g_c) \right\} = \sum_{p=1}^P 2^{(p-1)*} f_8(TLOP_p^n(g_c)) \left| \left\{ \bar{D} \mid \forall D, \exists -I_D^{n-1}(g_c) \right\} \right. \tag{23}$$

For each of remaining 7 directions, an 8-bit pattern is achieved, that is merged with magnitude pattern. Magnitude pattern is calculated as:

$$MP = \sum_{p=1}^P 2^{n-1*} f_9(M_{I(g_p)} - M_{I(g_c)}) \tag{24}$$

where

$$M_{I(g_p)} = \sqrt{\sum \left( \left( I_{0^0}^{n-1}(g_p) \right)^2 + \left( I_{45^0}^{n-1}(g_p) \right)^2 + \left( I_{90^0}^{n-1}(g_p) \right)^2 + \left( I_{135^0}^{n-1}(g_p) \right)^2 \right)} \tag{25}$$

$$f_9(x) = \begin{cases} 1, & x \geq 0 \\ 0, & else \end{cases} \tag{26}$$

Figure 6 illustrates the possible Local Octa Patterns for a center pixel (shown in red color) in a particular segment of an image with neighbors (shown in green color). D(c), D(1), D(2) are the directions of center, first neighbor, and second neighbor pixels; Whereas, M(c), M(1), M(2) are magnitudes of center, first neighbor, and second neighbor pixels and so on. If direction of neighbor pixel is that of center pixel then T-LOP bit is taken as 0 otherwise it is same as the direction of neighbor pixel. So the T-LOP pattern comes out to be 41,383,183. This pattern is then subdivided into seven binary patterns; first pattern is achieved by replacing 2 with 1 and all other values with 0 in T-LOP, second pattern is achieved by replacing 3 with 1 and other values with 0 and so on. Magnitude pattern is achieved by comparing magnitude values of neighbor pixels with center pixel and pattern achieved in this case is 11,100,001. These eight patterns are used to describe the texture of an image.

Higher order TLOP can also be used to describe texture, which extracts more information but it has been found in simulations that second order patterns give best results. Moreover, in the pictorial explanation, nearest neighbors are considered for pattern calculation. Neighbors at the diagonal 5 and 7 from the center pixel can also be considered for pattern calculation thus giving name Tetragonal Octa Patterns. Figure 7 shows neighbors at different diagonal locations; Red represents center pixel; yellow pixels are nearest neighbors at diagonal 3; green pixels are neighbor pixels at diagonal 5; blue pixels are neighbors at diagonal 7; whereas pink pixels represent horizontal, vertical and diagonal pixels of the blue cornered pixel.

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(c)=6  
M(c)=6.164

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(1)=4  
M(1)=7.681

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(2)=1  
M(2)=8.944

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(3)=3  
M(3)=10

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(4)=8  
M(4)=4.899

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(5)=3  
M(5)=4.243

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(6)=1  
M(6)=5.477

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(7)=8  
M(7)=4.359

2	5	3	9	1
6	7	9	1	5
3	8	4	2	9
2	3	5	8	2
1	2	3	2	1

D(8)=3  
M(8)=8.544

**Octa Pattern = 41383183**

**Pattern1 = 00000000, Pattern2 = 00101001, Pattern3 = 10000000,**

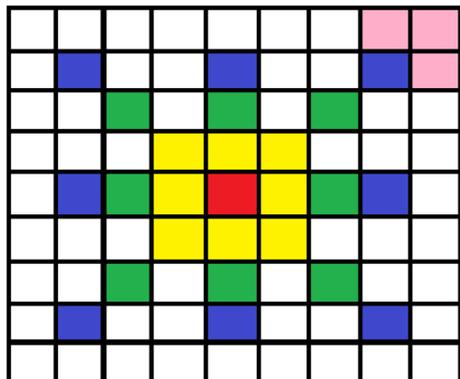
**Pattern4 = 00000000, Pattern5 = 00000000, Pattern6 = 00000000,**

**Pattern7 = 00010010; Magnitude Pattern = 11100001**

Fig. 6 Local Octa Pattern calculation

Concatenating histograms of each pattern makes feature vector. In order to reduce the dimensions of feature vector for our experiments, reduced histogram of each pattern has been used. Where each pattern histogram is of 20 bins making the feature vector of length 160.

Fig. 7 Neighbors at different diagonal locations



## 4.2 Class misbalancing issue in image classification and its rectification

In classification, a class is said to be misbalanced if number of samples in that class are less than samples in other classes. In image classification where image classes are so closely related with each other, this misbalancing problem is highly observed. After classification, images of one class associate with other class images thus raising this misbalancing issue and affecting the retrieval precision. In the proposed Local Octa Pattern algorithm, the target was to achieve more elaborated feature vectors of images. The idea of reducing misbalancing problem has been materialized by increasing the positive samples of a class. If we are able to increase positive samples in a particular class, we can train our classifier on these enhanced samples which will result in enhanced precision and reduced misbalancing issue, which is the main focus and optimization problem of this work.

If there are  $|D|$  images in an image database which correspond to  $n$  different classes then the whole image database can be divided into  $S = \{s_1, s_2, \dots, s_n\}$  subsets. Each subset will be having  $|D|_{s_j} = |D|/S \times 0.3$  images from each category, where 0.3 corresponds to 30%. Actually 30% images of each category have been used as training images.

For every subset there are two further sub-categories; positive  $|d|^+$  and negative  $|d|^-$  samples; written as in [17]:

$$|d|^+ = \left\{ j = 1, 2, \dots, |D_{s_j}| \mid I_j \in |D_{s_j}| \right\} \quad (27)$$

$$|d|^+ = \left\{ j = 1, 2, \dots, |D_{S-s_j}| \mid I_j \in |D_{S-s_j}| \right\} \quad (28)$$

These are the sets which are used for classifier training but unfortunately classifier can be easily misled due to misbalancing. Therefore a precise algorithm is needed which can deal with this issue; Genetic Algorithm resolves the purpose and is explained below:

Genetic Algorithm (GA) is based on principles of natural genetics. In GA, candidate solutions or decision variables to a search problem are known as *chromosomes*, chromosomes are made up of *genes*, and value of genes is known as *alleles*. A certain parent population, known as elite parents, undergoes an evaluation process; offspring formed in this process, which clear an evaluation test, also become the part of elite population set. GA generates multiple populations in this process and returns a population which is best suited or has the highest fitness when tested against a fitness function. Algorithm has been further explained below.

### 4.2.1 Structure of chromosomes and population generation

Chromosome can be defined as:

$$\Theta = [\theta_{c,1}, \theta_{c,2}, \dots, \theta_{c,G}], c = \{1, 2, \dots, P\} \quad (29)$$

where G is the number of genes in a chromosome and P represents population size. The task is to generate population of offspring. For this we have to select initial elite parent population. Initial population is the original images' feature vectors. So, in our case, feature vectors obtained after applying LOcP on original images, serve as parent population. Different genetic operators may be applied on these chromosomes in order to generate offspring population.

These genetic operators include mutation [1] and crossover [15]. Crossover may be further divided into mid-point crossover or 2-point crossover. In our approach, we have used 2-point crossover for offspring population generation. The procedure is explained below.

In this method, 2 random parents and cut-points are selected. Then the segments of the parents are exchanged to generate two new offspring. If  $\Theta_1$  and  $\Theta_2$  are the chromosomes of parents,  $\Theta_{1,off}$  and  $\Theta_{2,off}$  are the chromosomes of newly generated offspring and  $p1 = \{p1, 1, p1, 2, \dots, p1, G\}$ ,  $p2 = \{p2, 1, p2, 2, \dots, p2, G\}$  represent the two parents respectively, then new offspring can be formed using:

$$parents(p)[1, 2] = rand(2) \tag{30}$$

$$crossover(C)[1, 2] = rand(2) \tag{31}$$

$$\Theta_{1,off} = [\theta_{p1,1} \rightarrow \theta_{p1,C1}; \theta_{p2,C1} \rightarrow \theta_{p2,C2}; \theta_{p1,C2} \rightarrow \theta_{p1,G}] \tag{32}$$

$$\Theta_{2,off} = [\theta_{p2,1} \rightarrow \theta_{p2,C1}; \theta_{p1,C1} \rightarrow \theta_{p1,C2}; \theta_{p2,C2} \rightarrow \theta_{p2,G}] \tag{33}$$

where C represents crossover or cut points. 2 random parents and cut-points are selected then the parents segments are merged to form two new offspring as indicated in Eqs. (32, 33). If in random number generation, 2 parents are the same that is  $p1 = p2$  then again 2 random numbers are generated till we get 2 different parents. Same is the case for  $C1 = C2$ . These newly generated offspring are now ready to become the part of elite parent population, if they pass the evaluation test. This new elite population will then be used to generate new offspring for further iteration. The evaluation test for the offspring chromosomes is:

$$\Theta_e = \|\Theta_{off} < \Theta\| e = \{p + 1 \rightarrow P\} \tag{34}$$

where  $e$  stands for elite. This means only those offspring chromosomes will be the part of elite population which has less distance than any of the elite parents' chromosome. Hence, after multiple random parent selection and selecting only those offspring who pass evaluation test, a new elite population is generated. In our case, a constant population size of 150 was used.

### 4.2.2 Fitness function

We have used below given fitness function:

$$J(\Theta_j) = \left\| \operatorname{argmin} \left( F_{avg|d|+}^j - F_{avg\Theta_{off}}^j \right) \right\|^\beta \tag{35}$$

According to the fitness criteria, we are selecting only that particular population, from the number of populations generated, which has the minimum average distance as compared to the average distance of original positive training samples. Here  $\beta$  defines the number of populations generated. We used  $\beta = 10$  in our experiments. Previously

we were generating chromosomes population using parent population but here the population means complete set of elite population consisting of both elite parents and offspring.

The complete Genetic Algorithm has been summarized below:

---

#### Genetic Algorithm GA

---

**Input:** positive image sample set  $|d|^+$ , images in database  $|D|_S$ ,  $\beta$  (number of generated populations), population size “PS”, Genetic Algorithm method “Gm”

1. for  $j \leftarrow 1, \beta$  do
2.  $\Theta^+ \leftarrow$  Genetic Algorithm ( $|d|^+$ ) with elite parents  $|d|^+$
3. for  $k \leftarrow 1, PS/2$  do
4.  $[p1, p2] \leftarrow \text{rand}(2)$
5.  $[C1, C2] \leftarrow \text{rand}(2)$
6.  $\Theta_{1,off} = [\theta_{PS1,1} \rightarrow \theta_{PS1,C1}; \theta_{PS2,C1} \rightarrow \theta_{PS2,C2}; \theta_{PS1,C2} \rightarrow \theta_{PS1,G}]$
7.  $\Theta_{2,off} = [\theta_{PS2,1} \rightarrow \theta_{PS2,C1}; \theta_{PS1,C1} \rightarrow \theta_{PS1,C2}; \theta_{PS2,C2} \rightarrow \theta_{PS2,G}]$
8. if Gm = elitism asymmetric then
9. compute  $J(\Theta_j) = \left\| \arg \min(F_{|d|^+}^j - F_{\Theta_{off}}^j) \right\|^\beta$
10. else
11. compute  $\Theta_e = \|\Theta_{off} < \Theta\|, e = \{1 \rightarrow P\}$
12. compute  $J(\Theta_j) = \left\| \arg \min(F_{|d|^+}^j - F_{\Theta_{off}}^j) \right\|^\beta$
13. end if
14. end for
15. end for

**Output:**  $|d|^+ = |d|^+ + \Theta_{off}$

---

### 4.3 Classification using support vector machines (SVM) and image retrieval

In our experiment, after generating new feature sets from Genetic Algorithm, SVM classifier [14] was trained on these features. All positive feature vectors belonging to a specific category were labeled with “1” while others were labeled with “0”. In this way classifier was trained on all categories of images. Therefore, when the problem of class association for the query image rises; only those images which have same class are returned to the user based on the distance *w.r.t* query image.

## 5 Experimental results

This section discusses about databases used, precision/Recall and experiments performed. In order to prove the effectiveness of our algorithm, several experiments have been performed.

### 5.1 Databases used

For making a retrieval system, first of all a suitable database is selected. There is no standard of image database, number of images and type of images in database. We have used COREL image database set A, Oxford Flowers Dataset, subsets of Caltech-101 and Caltech-256 for experimental purposes. Databases are easily available and

have been used by most researchers. COREL database contains a wide variety of images. Database has 1000 images which are divided into 10 classes; African people, beach, building, buses, dinosaurs, elephants, flowers, horses, mountains and food [22]. Each class has 100 images with resolution of either  $384 \times 256$  or  $256 \times 384$ . Oxford Flowers dataset has 1360 images divided into 17 flower categories; each category has 80 images. Images are of varying resolution and dataset was downloaded from [31]. 10 categories for each of Caltech-101 and Caltech-256 were used for experiment. Categories used for Caltech-101 are accordion, airplanes, anchor, ant, barrel, bass, beaver, binocular, bonsai and brain while categories used for Caltech-256 dataset are ak-47, american flag, backpack, baseball-bat, baseball-glove, basketball-hoop, bat, bathtub, beer-mug and blimp. Caltech-101 dataset was downloaded from [3] while Caltech-256 was downloaded from [4].

## 5.2 Retrieval precision and recall rates

To evaluate the performance of retrieval system, we have to determine how many relevant results are achieved when query image is fed to the system. Performance metric used in this research is precision/recall rates. Precision rate is defined below:

$$P = \frac{R_{ret}}{T_{ret}} \quad (36)$$

where P is Precision rate,  $R_{ret}$  is no. of relevant retrieved images and  $T_{ret}$  is total no. of retrieved images. While Recall rate is defined as:

$$R = \frac{R_{ret}}{T_{rel}} \quad (37)$$

where R is Recall rate,  $R_{ret}$  is number of relevant images retrieved and  $T_{rel}$  is total number of relevant images.

Results have been calculated after 5 iterations on randomly selecting query images from each category. 5 query images were randomly selected from each category. Top 20 retrieved images were used to calculate precision and recall rates.

## 5.3 Retrieval results

We have performed different experiments. First of all Local Octa Patterns (2nd order) with neighbors at diagonal 3, 5 and 7 were used. Precision rates observed in this case have been shown in Fig. 8. COREL Database has been selected initially. It can be seen easily when neighbors at diagonal ( $D$ )=3 were chosen, highest average precision was observed. Then a second set of experiments were performed using 2nd, 3rd and 4th order LOcP. Neighbors at diagonal=3 were considered in this case. Precision results have been shown in Fig. 9 and it is obvious from the graph that 2nd order LOcP produces most efficient results. So, the results depict, enhanced efficiency has been achieved when 2nd order LOcPs are used while considering neighbors at diagonal=3.

Our proposed method results have been compared with those of some previous techniques and improved precision was observed. Comparison of Average Retrieval Rate (ARR) has been made in Table 1 below. We have taken 2nd order LOcPs and neighbors at diagonal=3 in

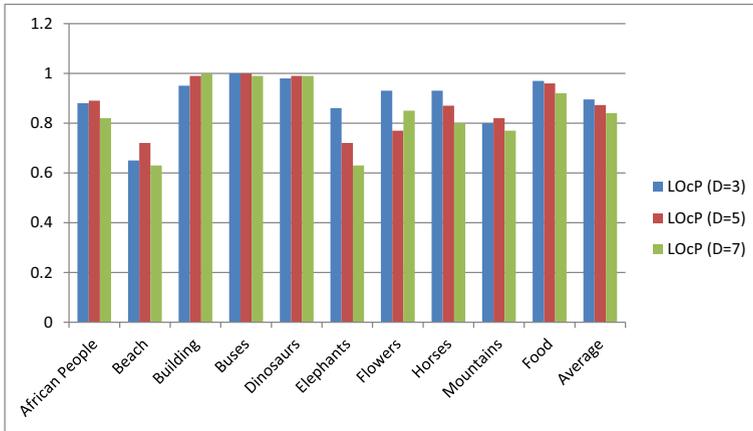


Fig. 8 Precision while considering neighbors at different diagonal locations for COREL dataset

proposed algorithm for comparison. Same parameters have been used for comparison in next experiments for other datasets.

In order to verify that GA in our proposed approach increases the retrieval precision, we have also implemented the approach without using GA. The results in both cases have been shown in Fig. 10.

Comparison of recall rates of proposed approach with those of previous approaches has been tabulated in Table 2. Average retrieval precision has also been observed for increased number of retrieved images e.g. 30, 40, 50 and so on. Behavior observed in this case has been shown in Fig. 11.

In our 2nd database experiment, Oxford Flowers Dataset has been used. When proposed algorithm was compared with other algorithms, improved results were observed. Comparison results have been shown in Fig. 12.

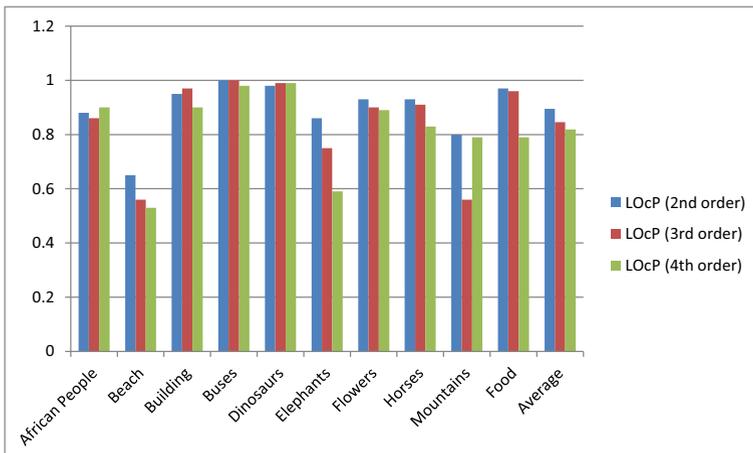
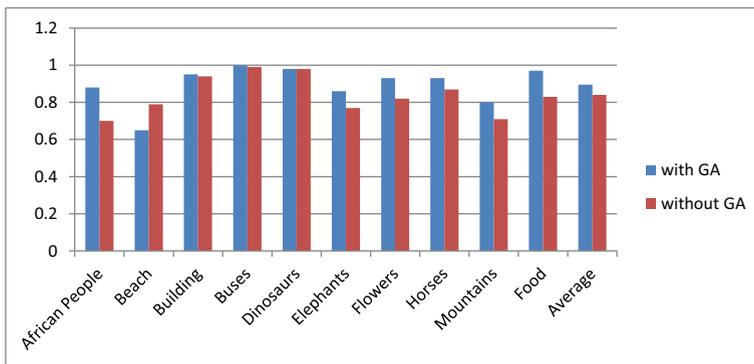


Fig. 9 Precision calculation using 2nd, 3rd and 4th order LOcPs for COREL dataset

**Table 1** ARR Comparison of Different image retrieval techniques for COREL Dataset

Category	ODBTC Indexing [12]	CCM + DBPSP [8]	Curvelet+Wavelet+ Gabor [17]	ICTEDCT-CBIR [38]	SURF+ HOG [27]	Proposed
African people	0.85	0.72	0.73	0.63	0.65	0.88
Beach	0.47	0.59	0.72	0.64	0.63	0.65
Building	0.68	0.59	0.79	0.69	0.68	0.95
Buses	0.88	0.89	1.00	0.91	0.90	1.00
Dinosaurs	0.99	0.99	0.97	0.99	1.00	0.98
Elephants	0.73	0.70	0.75	0.78	0.79	0.86
Flowers	0.96	0.93	0.86	0.94	0.96	0.93
Horses	0.94	0.86	0.82	0.95	0.93	0.93
Mountains	0.47	0.56	0.69	0.73	0.67	0.80
Food	0.81	0.77	0.90	0.80	0.80	0.97
Average Precision	0.78	0.76	0.82	0.81	0.80	0.89

**Fig. 10** Average retrieval precision with and without using GA

In 3rd and 4th experiments, subsets of Caltech-101 and Caltech-256 were used. Comparison of proposed approach with previous approaches in this case has been tabulated in Tables 3 and 4. More precise results in proposed approach case can be observed from the tables.

**Table 2** Recall Comparison of Different image retrieval techniques for COREL Dataset

Category	ODBTC Indexing [12]	CM + DBPSP [8]	Curvelet+Wavelet+ Gabor [17]	ICTEDCT-CBIR [38]	SURF+ HOG [27]	Proposed
African people	0.17	0.16	0.15	0.13	0.15	0.18
Beach	0.09	0.20	0.14	0.13	0.15	0.13
Building	0.14	0.19	0.16	0.14	0.16	0.19
Buses	0.18	0.13	0.20	0.18	0.19	0.20
Dinosaurs	0.20	0.11	0.19	0.20	0.19	0.20
Elephants	0.15	0.16	0.15	0.16	0.17	0.17
Flowers	0.19	0.13	0.17	0.19	0.17	0.19
Horses	0.19	0.14	0.16	0.19	0.17	0.19
Mountains	0.09	0.24	0.14	0.15	0.17	0.16
Food	0.16	0.15	0.18	0.16	0.16	0.19
Average Recall	0.156	0.16	0.165	0.163	0.17	0.18

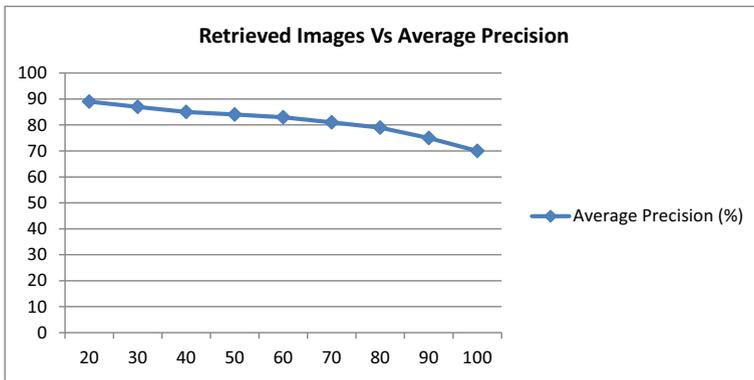


Fig. 11 Top retrieved images vs average precision for COREL dataset

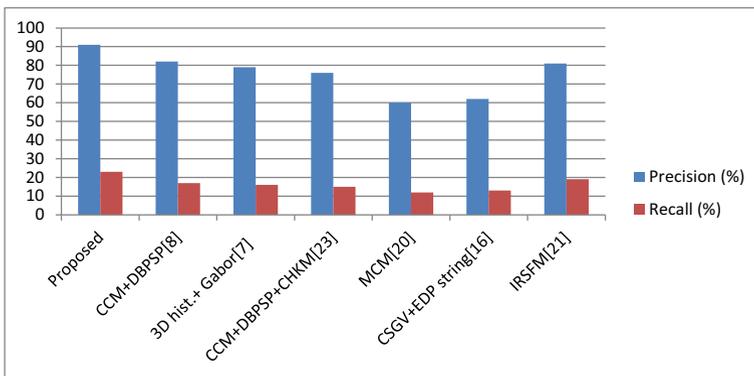


Fig. 12 Comparison of proposed approach with other methods for flowers dataset

Table 3 ARR comparison of different approaches for Caltech-101 dataset

Method	Caltech-101				
	SIFT <sub>BoW</sub> [33]	IFFS <sub>BoW</sub> [33]	IFFS <sub>Tree</sub> [33]	IFFS <sub>GBR</sub> [33]	Proposed
ARR(%)	74	66	60	68	84.6

Table 4 ARR comparison of different approaches for Caltech-256 dataset

Method	MN-ARM [37]	MN-IRM [37]	SIS [37]	SIMPLicity [37]	MN-MIN [37]	Proposed
ARR (%)	22.5	22	21.7	21.5	21.3	31.1

## 6 Conclusion

In this paper we have proposed a content based image retrieval technique for efficiently retrieving images from a database. The target was to achieve a more accurate algorithm so LOcPs have been used for feature extraction. In LOcPs, we have used diagonal derivative in addition to horizontal and vertical derivatives to get the direction of center pixel. This direction is then used to calculate improved pattern, and the magnitude pattern has been achieved using magnitudes of derivatives. We have used different order LOcPs; and neighbors at different diagonal locations were considered, and found that LOcPs with 2nd order having neighbors at diagonal = 3 produce the most efficient results. Genetic Algorithm has been further applied on these feature vectors to resolve the class miss balancing problem. The algorithm has been found more effective than other binary patterns. Proposed algorithm was tested on four different databases and improved precision and recall results were observed in all cases.

## References

1. Arevalillo-Herr'aez M, Ferri FJ, Moreno-Picot S (2011) Distance-based relevance feedback using a hybrid inter active genetic algorithm for image retrieval. *Appl Soft Comput* 11(2):1782–1791
2. Bay H, Ess A, Tuytelaars T, Gool LV (2008) Speeded-up robust features. *Comput Vis Image Underst* 110(3):346–359
3. Caltech-101 dataset. Online available on: [https://www.vision.caltech.edu/Image\\_Datasets/Caltech101](https://www.vision.caltech.edu/Image_Datasets/Caltech101). Accessed 13 Dec 2016
4. Caltech-256 dataset. Online available on: [http://www.vision.caltech.edu/Image\\_Datasets/Caltech256](http://www.vision.caltech.edu/Image_Datasets/Caltech256). Accessed 13 Dec 2016
5. Campana BJL, Keogh EJ (2010) A compression-based distance measure for texture. *Statist Anal Data Mining* 3(6):381–398
6. Cinque L, Ciocca G, Levialdi S, Pellicano A, Schettini R (2001) Color based image retrieval using spatial chromatic histograms. *Image Vis Comput* 19(13):979–986
7. ElAlami ME (2011) A novel image retrieval model based on the most relevant features. *Knowl-Based Syst* 24:23–32
8. ElAlami ME (2013) New matching strategy for content based image retrieval system. *Appl Soft Comput* 14: 407–418
9. Freeman WT, Adelson EH (1991) The design and use of steerable filters. *IEEE Trans Pattern Anal Mach Intell* 13(9):891–906
10. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C Appl Rev* 42(4):463–484
11. Guha T, Ward RK (2014) Image similarity using sparse representation and compression distance. *IEEE Trans Multimedia* 16(4):980–987
12. Guo JM, Prasetyo H (2015) Content-based image retrieval using features extracted from Halftoning based block truncation coding. *IEEE Trans Image Process* 24(3):1010–1024
13. Harris C, Stephens M (1988) A combined corner and edge detector. In: *Alvey vision Conf.*, pp 147–151
14. Hearst MA et al (1998) Support vector machines. *IEEE Intelligent Systems and their Applications* 13(4):18–28
15. Herrera F, Lozano M, Sanchez A (2005) Hybrid crossover operators for real-coded genetic algorithms : an experimental study. *Soft Comput* 9:280–298
16. Huang PW, Dai SK (2003) Image retrieval by texture similarity. *Pattern Recogn* 36(3):665–679
17. Irtaza A, Jaffar MA (2014) Categorical image retrieval through genetically optimized support vector machines (GOSVM) and hybrid texture features. *SIVIP* 9(7):1503–1519
18. Irtaza A, Jaffar MA et al (2014) Embedding neural networks for semantic association in content based image retrieval. *Multimed Tools Appl* 72(2):1911–1931
19. Jain AK, Vailaya A (1996) Image retrieval using color and shape. *Pattern Recogn* 29(8):1,233–1,244
20. Jhanwar N, Chaudhuri S, Seetharaman G, Zavidovique B (2004) Content based image retrieval using motif co-occurrence matrix. *Image Vis Comput* 22:1211–1220
21. Jin C, Ke S-W (2017) Content-based image retrieval based on shape similarity calculation. *3D Res* 8:23

22. Lai C-C, Chen Y-C (2011) A user-oriented image retrieval system based on interactive genetic algorithm. *IEEE Trans Instrum Meas* 60:3318–3325
23. Lin C-H, Chen R-T, Chan Y-K (2009) A smart content-based image retrieval system based on color and texture feature. *Image Vis Comput* 27:658–665
24. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
25. Manjunath BS, Ma W-Y (1996) Texture features for browsing and retrieval of image data. *IEEE Trans Pattern Anal Mach Intell* 18(8):837–842
26. Mao J, Jain AK (1992) Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recogn* 25(2):173–188
27. Mehmood Z, Abbas F, Mahmood T, Javid MA, Rehman A, Nawaz T (2018) Content-based image retrieval based on visual words fusion versus features fusion of local and global features. *Arab J Sci Eng*:1–20
28. Murala S, Maheshwari RP, Balasubramanian R (2012) Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Trans Image Process* 21(5):2874–2886
29. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987
30. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int J Comput Vis* 42(3):145–175
31. Oxford Flowers Dataset. Online available on: <http://www.robots.ox.ac.uk/~vgg/data/flowers/17/>. Accessed 13 Dec 2016
32. Pass G, Zabih R, Miller J (1997) Comparing images using color coherence vectors. In: Proceedings of the fourth ACM international conference on multimedia. <https://doi.org/10.1145/244130.244148>
33. Raveaux R, Burie JC, Ogier JM (2013) Structured representations in a content based image retrieval context. *J Vis Commun Image Represent* 24:1252–1268
34. Swain MJ, Ballard DH (1991) Color indexing. *Int J Comput Vis* 7(1):11–32
35. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans Image Process* 19(6):1635–1650
36. Wang JZ, Li J, Wiederhold G (2001) SIMPLcity: semantics-sensitive integrated matching for picture libraries. *IEEE Trans Pattern Anal Mach Intell* 23(9):947–963
37. Yang X, Cai L (2014) Adaptive region matching for region-based image retrieval by constructing region importance index. *IET Comput Vis* 8(2):141–151
38. Youssef SM (2012) ICTEDCT-CBIR integrating curvelet transform with enhanced dominant colors extraction and texture analysis for efficient content based image retrieval. *Comput Electr Eng* 38(5):1358–1376
39. Zhang B, Gao Y, Zhao S, Liu J (2010) Local derivative pattern versus local binary pattern: face recognition with higher-order local pattern descriptor. *IEEE Trans Image Process* 19(2):533–544

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Zain Shabbir** received his BSc and MS degrees in Electrical Engineering both from University of Engineering and Technology (UET) Taxila. Currently he is serving in Electrical Department of UET Lahore-Faisalabad campus as Lecturer. His research interests include Machine learning and Image retrieval.



**Aun Irtaza** received his M.S. degree in computer science in 2008 from National University of Computer and Emerging Sciences, NU-FAST, Islamabad, Pakistan. He also received his Ph.D. degree from Department of Computer Science, NU-FAST. His research interests include image processing and computational intelligence.



**Ali Javed** received his MS and Ph.D degrees in Computer Engineering both from University of Engineering and Technology (UET), Taxila, Pakistan. Currently he is serving as an Assistant professor in Software Engineering Department, UET Taxila. His research interests include Object detection, tracking and recognition, video surveillance and summarization.



**Muhammad Tariq Mahmood** received the M.C.S degree in computer science from AJK University of Muzaffarabad Pakistan in 2004 and the M.S. degree in intelligent software systems from Blekinge Institute of Technology Sweden in 2006 and the Ph.D. degree in school of information and mechatronics from Gwangju Institute of Science and Technology, Gwangju Korea in 2011. He is currently a Full-Time Lecturer in the School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, Korea. His research interests include image processing, 3-D shape recovery from image focus, computer vision, pattern recognition, and machine learning.