Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-08401-7

An efficient bi-layer content based image retrieval system

Sachendra Singh[1] · **Shalini Batra**[1]

Received: 16 November 2018 / Revised: 17 July 2019 / Accepted: 17 October 2019 /
Published online: 21 February 2020

Abstract
Large amount of multi-media content, generated by various image capturing devices, is shared and downloaded by millions of users across the globe, every second. High computation cost is inured in providing visually similar results to the user's query. Annotation based image retrieval is not efficient since annotations vary in terms of languages while pixel wise matching of images is not preferred since the orientation, scale, image capturing style, angle, storage pattern *etc.* bring huge amount of variations in the images. Content Based Image Retrieval (CBIR) system is frequently used in such cases since it computes similarity between query image and images of reference dataset efficiently. A Bi-layer Content Based Image Retrieval (BiCBIR) system has been proposed in this paper which consists of two modules: first module extracts the features of dataset images in terms of color, texture and shape. Second module consists of two layers: initially all images are compared with query image for shape and texture feature space and indexes of M most similar images to the query image are retrieved. Next, M images retrieved from previous layer are matched with query image for shape and color feature space and F images similar to the query image are returned as a output. Experimental results show that BiCBIR system outperforms the available state-of-the-art image retrieval systems.

Keywords Content based image retrieval · Feature space · Sub-space features · Layer based image retrieval

1 Introduction

Frequently used image capturing devices like smart phones, cameras, closed circuit television camera, *etc.* generate large volumes of visual data. Further, millions of images are

✉ Sachendra Singh
sachendrac@gmail.com

Shalini Batra
sbatra@thapar.edu

[1] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India

uploaded on the social networking websites in a single day [49]. This huge amount of visual data uploaded by users from various geographic regions with varied languages have either meta data in diverse languages or no meta data associated with the images. To identify similar type of images from such unstructured visual data is a challenging task. Content Based Image Retrieval (CBIR) system can enhance the similar image search capability, especially for images having multilingual tagging and annotations.

Image retrieval approach used in CBIR system is quite different from the image retrieval systems which use meta data based approach for images retrieval. In CBIR, initially image space is converted to feature space. Features are extracted in the form of color, texture, shape, *etc.* which are represented in the form of a feature vector. Image similarity is computed using feature vectors of query image and dataset images through a distance measure function and images most similar to the query image are returned as output. A general overview of CBIR system is given in Fig. 1.

CBIR systems can be used in wide variety of applications such as bio-metric system, digital library, multimedia recommender system [3], 3D object retrieval [9], multimedia event detection [2, 16], *etc.* Google images, TinEye, eBay, SK Planet, Flipkart, *etc.* use CBIR based similarity search. These websites help users in searching the desired images by uploading or selecting an image from the given set of images. To retrieve the visually similar images based on image descriptors many similarity search based approaches have been proposed [23, 30, 41].

CBIR systems work in two stages: In the first stage called indexing; features of the dataset images are extracted and stored in feature vectors. In second stage, query image features are compared with images in the dataset. Since the image retrieval techniques based

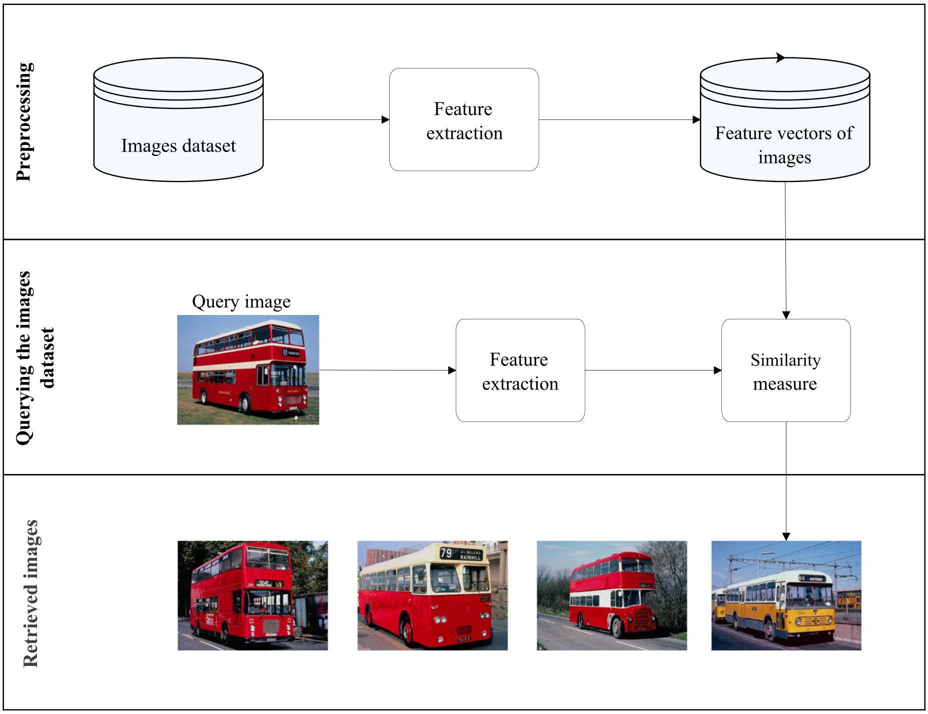

Fig. 1 Overview of general CBIR model

on single image feature are sub-optimal, focus has now shifted towards the use of multiple image features to represent images effectively. The primitive image features which include shape, texture, color and spatial information are best suited for image retrieval from versatile image datasets. Many CBIR systems have been proposed which use multiple features to represent an image [1, 41].

When the image dataset is quite large, retrieval of similar images requires huge amount of time. Further, in massive image dataset, images relevant to query image are very few. Hence, it is important to clean those irrelevant images. To minimize the computational cost, various image retrieval systems have been introduced which use layered approach to filter irrelevant images. In this work an efficient bi-layer content based image retrieval (BiCBIR) system is proposed which uses first layer for image filtering and second layer for retrieval of images similar to query image. The major contributions of this work are:

1. An efficient bi-layer content based image retrieval system has been proposed which uses one common feature in both layers. In first stage, similarity of all images in the dataset is computed with query image on some feature space and M most similar images are passed to second layer. In the second layer, M images are compared with query images and F similar images are returned to the user.
2. The proposed system reduces the total number of comparisons without compromising the accuracy.
3. Proposed CBIR system has been tested on COREL and GHIM20 datasets and results achieved outperform the existing state-of-the-art techniques.

The rest of this article is organized as follows: Section 2 introduces few techniques related to CBIR studied in the literature. In Section 3, proposed approach is discussed. Section 4 provides implementation details and results of the proposed approach. Finally, paper concludes with few directions on future work in Section 5.

2 Related work

This section explores the various CBIR approaches related to the proposed work. CBIR system is mainly affected by image feature selection and extraction technique, number of features used to represent the image, similarity measure methods used to compute the similarity between two images and retrieval methods used for searching images from the image dataset. Major concern in CBIR systems is that an image can have many versions which differ in size or color and it can be seen from different view points, thus making it hard to compare the images pixel by pixel. To avoid dissimilarity of similar images due to rotation, translation, scaling, *etc*., features of the images are extracted and matched to compute the similarity among images [22]. Several image retrieval systems have been developed which retrieve similar images on the basis of image features also known as image contents [14, 18].

CBIR systems use low level features of images which include color, texture, shape *etc.*, to retrieve images from the dataset. These features can be global or local. Global features describe an image as a whole and can be interpreted as a particular property of the image involving all pixels, whereas local features detect key points or region of interest in an image to describe the same. Global features have fast retrieval rate but they are not effective in comparing region of interest [41, 46]. Local features [29, 33] of images are invariant to translation, rotation, scale, *etc*. An important concern with CBIR systems is the semantic gaps [14, 40] among high-level image concepts and low-level image features. The Bag-of-Visual-Words (BoVW) model is a standard approach to map local features into a vector of fixed length [45]. Feature vectors are quantized into visual words formulated by clustering the image features [15]. BoVW is widely used image feature representation method [37] in many applications of computer vision [17]. Pavithra et al. [24] developed a hybrid framework for CBIR system to address the accuracy issues associated with the traditional image retrieval systems. This framework initially selects pertinent images from a large database using color moment information. Some CBIR systems use multiple layers for feature matching and each layer compares only one feature [26, 30]. Pradhan et al. [26] proposed an hierarchical CBIR framework which uses adaptive tetrolet transform for texture feature extraction. Color and shape features are extracted by using color channel correlation histogram and edge joint histogram respectively. This framework works in three phases and matches one feature in each phase. Images similar to query image are passed to next stage for subsequent feature matching. The main advantage of this approach is that it compares only relevant images after first stage and discard non-relevant images in every upcoming stage, reducing the number of images to be compared in subsequent stages. Due to orderless histogram generated by Bag of Visual Word (BoVW), spatial contents are ignored. To overcome the problem of BoVW, Mehmood et al. [20] proposed a weighted average of triangular histogram (WATH) for image representation. The WATH includes the spatial contents in the inverted index of BoVW model. To avoid high computation involved in image segmentation, Jian et al. [12] introduced a perception based directional patch extraction and salient patch detection method to extract local features for CBIR. Zhou et al. [50] merged color histogram, local directional pattern and dense Scale Invariant Feature Transform (SIFT) features to represent the images more accurately. Fadaei et al. [7] proposed a CBIR scheme which uses color and texture features for image representation. In this scheme, uniform partitioning is applied on HSV color space to extract Dominant Color Descriptor (DCD) features. Texture features are extracted by using wavelet and curvelet to avoid noise and translation problems associated with image retrieval. Color and texture features are combined by assigning optimal weights with the help of particle swarm optimization. Yue et al. [47] designed a CBIR system which uses color histogram for color feature and co-occurrence matrix for texture feature extraction. Further, weights are assigned to color and texture feature vectors for comparing query and dataset images. Recently, Cheng et al. [3] proposed a multi-model aspect-aware topic model for recommender systems which uses text reviews and item images.

Yildizer et al. [44] proposed an ensemble approach which uses multiple support vector machines. Feature vectors of images are generated by using Daubechies wavelet transform. Choraś et al. [27] introduced a CBIR methodology which uses combination of color, texture and shape features for similarity computation. In this scheme, color features are extracted using histograms of color moments in YUV space. Texture features are extracted using thresholded Gabor filter and shape features are generated using Zernike moments. Phadikar et al. [25] proposed a scheme which works in compressed form. Images are compressed using discrete cosine transform and features of the images are directly extracted from compressed dataset. Image features are in the form of color moment, color histogram and edge histogram. Further, genetic algorithms have been used to assign the optimal weights to all the feature types and Euclidean distance is used to compute similarity between query image features and dataset images. Zhu et al. [51] proposed a CBIR system which consists of three steps: a) preprocessing which uses manifold to prune out the irrelevant images, b) similarity between query image and images remaining after pruning, which is computed by utilizing the probability density estimation, c) a random walk with restart model (RWRM), which is used to refine the ranking between query image and unlabeled images.

Tong et al. [35] proposed an algorithm for image retrieval based on granular computing. The advantage of this approach is that it also work in case of disordered image information. Shao et al. [28] introduced supervised two-stage deep learning cross-modal retrieval which supports text to image and image to text retrieval. Hashing based image retrieval

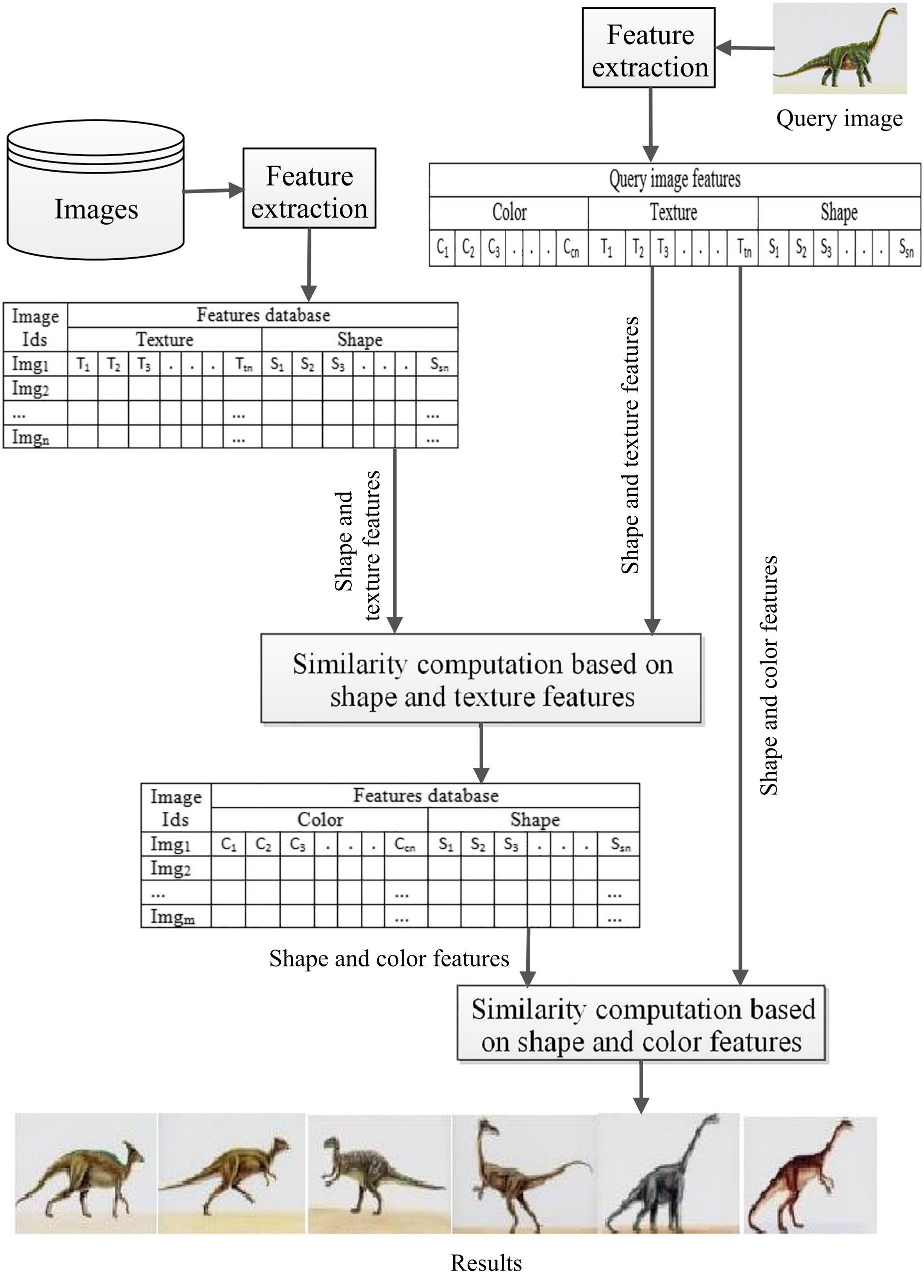

Fig. 2 Block diagram of proposed Bi-layer CBIR system

systems are faster as compared to traditional CBIR systems but they are sub-standard in terms of accuracy. To overcome the accuracy issues Wang et al. [38] proposed a variational Bayes framework to learn robust hash code which provide better accuracy. Yan et al. [43] introduced a hashing based image retrieval method which utilize deep learning network to generate hash code for images. Garcia and Vogiatzis [10] introduced a non-metric similarity computation method based on neural network. Deep learning based image features have been recently proposed by Wang et al. [39] in which top layer image features are used in CBIR instead of intermediate layer image features leading to fast retrieval with high accuracy. Heidy et al. [19] proposed an image annotation scheme, based on chain classifiers, which employs ensemble approach for classifier in the supervised image annotation. Each model in the chain deals with the same classification problem, making the proposed method an ensemble model build from multi-modal data. Particle Swarm Optimization (PSO) algorithms have been used for feature weighting [31] and clustering [42] to enhanced the performance of CBIR system. Mezzoudj et al. [21] proposed a parallel k-NN search for CBIR system which uses Spark and MapReduce to speedup the indexing and searching process.

Existing CBIR systems, studied in the literature, require multiple scan of entire dataset for query image retrieval and number of scans depend on the number of feature spaces used for image representation. There are some CBIR systems [26, 30] which avoid exhaustive search by searching the entire dataset in three stages. In first stage, single feature is used to prune non-relevant images based on that particular feature space. In second stage, feature different from previous stage, is used to prune out non-relevant images and finally third feature is used to extract relevant images. In such layered approaches, images filtered after first stage of pruning are passed as input to second layer. Since different feature space is used in second layer, similarity achieved in the first stage is lost and hence, some of relevant images may be dropped in second layer. Similarly, at third layer, images which are relevant for second layer but not for third layer feature space are dropped. To overcome this issue, proposed BiCBIR system uses two layers for image retrieval and same type of features for image representation.

3 Proposed CBIR model

In this work, an Efficient Bi-layer Content Based Image Retrieval System named as BiCBIR has been proposed where three primitive image features namely color, texture and shape have been considered. The similarity between query image, Q_i and dataset images, I_{DB} is computed in two layers. The proposed BiCBIR system is divided into two modules. In the first module, image features of the dataset images are extracted in the form of color, texture and shape (Section 3.1). Second module performs the retrieval task which is further divided into two layers (Section 3.2). The overview of the proposed system is demonstrated in Fig. 2.

3.1 Feature extraction

In this module, features are extracted from the image dataset $I_{DB} = \{I_1, I_2, I_3, \ldots, I_n\}$, consisting of n images and features considered in BiCBIR for a particular image I are: color I^c, texture I^t and shape I^s along with fixed size feature vector (Fig. 3); details are provided in Table 1. Algorithm 1 is used to extract features from image dataset and store into feature database (Table 2).

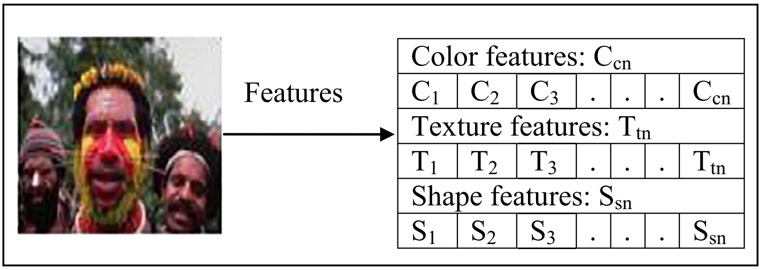

Fig. 3 Vector representation of extracted image features

For a image I, separate feature vector is created for color features IF^c, texture features IF^t and shape features IF^s.

Algorithm 1 Image features extraction.

Input:
I_{DB} : COREL and GHIM image datasets
$C_{H=9}$: Hue of HSV color space for an image
$C_{S=3}$: Saturation of HSV color space for an image
$C_{V=3}$: Value of HSV color space for an image
$T_{S=5}$: Scales of image for texture features
$T_{O=6}$: Orientations of image for texture features
$S_{M=5}$: Repeatition of moments of an image for shape features
Output:
F_{DB}: Features database of I_{DB} which includes following vectors
C_f : Color feature vectors having dimensions $n \times (9 \times 3 \times 3)$ 81
T_f : Texture feature vectors having dimensions $n \times (5 \times 6 + 5 \times 6)$ 60
S_f : Shape feature vectors having dimensions $n \times 21$
where n is number of images in dataset
imgCount=size(I_{DB})
for *i=1 to imgCount* **do**
$C_f[i]$ = colorFeature($I_{DB}[i]$, C_H, C_S, C_V)
$T_f[i]$ = textureFeature($I_{DB}[i]$, T_S, T_O)
$S_f[i]$ = shapeFeature($I_{DB}[i]$, M)
$I_{imagePath} = I_{DB}[i]$
$F_{DB}[i] = [C_f[i]T_f][i]S_f[i]]$
end
return F_{DB}

Table 1 Description of image feature type and vector length

Type	Vector length	Description
Color	81	Color features are extracted by using histogram on HSV color space
Texture	60	Texture features are obtained from Gabor filter with five scales and six orientations
Shape	21	Shape features are generated obtained from zernike moments

Table 2 Structure of image feature vectors in database

Image Ids	Features database																				
	Color							Texture							Shape						
Img_1	C_1	C_2	C_3	.	.	.	C_{cn}	T_1	T_2	T_3	.	.	.	T_{tn}	S_1	S_2	S_3	.	.	.	S_{sn}
Img_2																					
...							...							...							...
Img_n							...							...							...

3.1.1 Color feature extraction

Color features are extracted by using histogram of quantized values of color in Hue (H), Saturation (S) and Value (V) color space. HSV color space is more robust to human perception as compared to RGB color space [32]. Due to robustness of HSV color space, first RGB images are converted to HSV color space and then uniform quantization is applied (1). Feature vectors are generated by considering the values of H=9, S=3 and V=3 to form the feature vector of size 81 bin. Representation of color feature vector of an image is given in (2).

$$H = \begin{cases} 0 & h \in [1, 40] \\ 1 & h \in [41, 80] \\ 2 & h \in [81, 120] \\ 3 & h \in [121, 160] \\ 4 & h \in [161, 200] \\ 5 & h \in [201, 240] \\ 6 & h \in [241, 280] \\ 7 & h \in [281, 320] \\ 8 & h \in [321, 360] \end{cases} \tag{1}$$

$$S = \begin{cases} 0 & s \in [0.00, 0.30] \\ 1 & s \in [0.31, 0.70] \\ 2 & s \in [0.71, 1.00] \end{cases} \quad V = \begin{cases} 0 & v \in [0.00, 0.30] \\ 1 & v \in [0.31, 0.70] \\ 2 & v \in [0.71, 1.00] \end{cases}$$

$$IF^c = \{IF_1^c, IF_2^c, IF_3^c, \ldots, IF_{cn}^c\} \tag{2}$$

3.1.2 Texture feature extraction

Gabor filter, introduced by Gabor in 1946 [8], is one of the widely used filter for texture feature extraction. It is a Gaussian function modulated by complex sinusoidal of frequency and orientation [36]. In this work, texture features of an image are extracted by using five scales (s) and six orientations (o). The usage of multiple s and o makes the features rotation and scaling invariant on texture feature space. Different combination of s and o form the feature vector for texture space of length sixty, where first thirty values represent mean and next thirty values represent standard deviation of texture descriptors. To construct texture feature vector, a two dimensional Gabor function $G_f(x, y)$ and its Fourier transform $FT(u, v)$ is considered, represented as:

$$GF(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j C_f x\right) \tag{3}$$

$$FT(u, v) = \exp\left(\frac{1}{2}\left[\frac{(u - C_f)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right) \tag{4}$$

Where $\sigma_u = \dfrac{1}{2\pi\sigma_x}$, $\sigma_v = \dfrac{1}{2\pi\sigma_y}$ and C_f is a constant representing the center frequency of the filter bank having the highest frequency. A Gabor filter bank having a number of bandpass filters, with varying center frequencies, bandwidths and orientations is controlled by the parameters of Gabor wavelets. An input image, $I(x, y)$ when filtered by the set of Gabor wavelets $G_f(x, y)$ is given as:

$$R_{mn}(x, y) = \int I(x, y) GF^*_{mn}(x - x_1, y - y_1) dx_1 dy_1 \tag{5}$$

where $RF_{mn}(x, y)$ is the filter response at the spatial location (x, y); m=1,2,. . .,s is the number of scales and n=1,2,. . . ,o is the number of orientations. It is assumed that local image regions are spatially homogeneous and the mean and standard deviation of the magnitude of the filter responses are used to represent the region for matching purposes:

$$\mu_{mn} = \int \int \mid RF_{mn}(x, y) \mid dxdy \tag{6}$$

$$\sigma_{mn} = \sqrt{\int \int (\mid RF_{mn}(x, y) \mid - \mu_{mn})^2 dxdy} \tag{7}$$

A feature vector is constructed using μ_{mn} and σ_{mn} as feature components and texture feature descriptor are given as:

$$IF^t = \{IF^m_{11}, IF^m_{12}, \ldots, IF^m s \times o, IF^{sd}_{11}, IF^{sd}_{12}, \ldots, IF^{sd}_{s \times o}\} \tag{8}$$

3.1.3 Shape feature extraction

Shape features are extracted using Zernike Moments (ZM) [34]. ZM are rotation invariant and use zernike polynomials to form feature vector to represent an image based on shape features. ZMs are defined as the projections of f(x,y) on a class of polynomials, called Zernike polynomials. The complete set of Zernike polynomials is defined as:

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{jm\theta} \tag{9}$$

where $R_{nm}(\rho)$ are real-valued radial polynomials and (10) indicates the orthogonal property of $V^*_{nm}(\rho, \theta)$:

$$\int\int_{\substack{0 \le \rho \le 1 \\ 0 \le \theta \le 2\pi}} V^*_{nm}(\rho, \theta) V_{n'm'}(\rho, \theta) \rho d\rho d\theta = \frac{\pi}{n+1} \delta_{nn'} \delta_{mm'} \tag{10}$$

where $*$ denotes the complex conjugate and $\delta_{mm'}$ is

$$\delta_{nn'} = \begin{cases} 1, & \mathrm{n} = \mathrm{n}' \\ 0, & otherwise. \end{cases} \tag{11}$$

ZM of order n with repetition m for a continuous image function $f(x, y)$ over a unit disk is:

$$A_{nm} = \frac{n+1}{\pi} \int \int_{unit\ disk} V^*_{nm}(x, y) f(x, y) dxdy. \tag{12}$$

For digital images, the integrals can be replaced by summations

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V^*_{nm}(x, y), \qquad x^2 + y^2 \le 1 \tag{13}$$

In this work, twenty one initial zernike moments are used to represent the images; shape feature vector is represented in (14).

$$IF^s = \{IF^s_1, IF^s_2, IF^s_3, \ldots, IF^s_{sn}\} \tag{14}$$

3.2 Retrieval of similar images

Features of the query image Q_i, are extracted and a separate feature vector is formed for each feature type*i.e.* color, texture and shape. The query image features vectors are represented in (15). Images which are visually similar to Q_i are retrieved by using a two layer approach given in Algorithm 2.

$$
\begin{aligned}
Q^c &= \{Q_1^c, Q_2^c, Q_3^c, \ldots, Q_{cn}^c\} \\
Q^t &= \{Q_1^t, Q_2^t, Q_3^t, \ldots, Q_{tn}^t\} \\
Q^s &= \{Q_1^s, Q_2^s, Q_3^s, \ldots, Q_{sn}^s\}
\end{aligned} \tag{15}
$$

In the first layer, two features namely shape and texture are used to compute the similarity of Q_i and I_{DB}. The indexes of M most similar images, produced by pruning the non-relevant images based on similarity computed from first layer (named as I_{DBM}) serves as input to the second layer. In second layer, shape and color features of Q_i and I_{DBM} are matched and the indexes of F most similar images, I_{DBOut}, are retrieved as output.

Algorithm 2 Retrieval process of BiCBIR.

Input:
Q: Query image
F_{DB}: Features database of I_{DB} which includes following vectors
RI_{L1}: Total number of images to be retreived from layer 1.
Output:
Q^C:Color feature vector of query image
Q^T:Texture feature vector of query image
Q^S:Shape feature vector of query image
$Img_{indexes}$:indexes of relevent images after pruning of non-relevent images
Q^C = colorFeature(Q, C_H, C_S, C_V)
Q^T = textureFeature(Q, T_S, T_O)
Q^S = shapeFeature(Q, M)
for *i=1 to n* **do**
    $SS_{L1}(i, 1)$ = imgIndex(i)
    $SS_{L1}(i, 2) = \dfrac{sim(Q^S, F_{DB}^S[i]) + sim(Q^T, F_{DB}^T[i])}{2}$
end
SM_{L1} = Sort(SS_{L1}, Ascending)
$RI_{DB} = SM_{L1}(1 : RI_{L1})$
for *i=1 to* RI_{L1} **do**
    $SS_{L2}(i, 1) = SM_{L1}[i]$
    $SS_{L2}(i, 2) =$
    $\dfrac{sim(Q^S, F_{DB}^S[RI_{DB}[i]]) + sim(Q^C, F_{DB}^C[RI_{DB}[i]])}{2}$
end
SM_{L2} = Sort(SS_{L2},Ascending)
$Return\ SM_{L2}(1 : F)$

In Algorithm 2, both layers use one common feature *i.e.* shape which is used to partially preserve the similarity computed in first layer to the second layer. The selection of features sequence in first and second layer is based on the experimental analysis. Experiments have been performed for all possible sequences and the sequence which generate the best retrieval rate is considered in the proposed BiCBIR. Having a common feature in two layers makes system robust for cases where query image has same shape and texture but different color for same image.

Algorithm 3 Similarity computation method.

Input:
Q^f : Feature vector query image for f feature type, which can be color, texture or shape
I^f : Feature vector of an image from dataset for f feature
DM : Distance measure function, which can be cosine or euclidean.
Output:
SS : Similarity score between Q^f and I^f by using DM
if f='*Shape*' *or* f='*Texture*' **then**
    $SS = \sum_{i=1}^{|f|} \sqrt{\left(I^f - Q^f\right)^2}$
else
    $SS = \frac{\langle I^f \cdot Q^f \rangle}{\|I^f\| \|Q^f\|}$
Return SS

3.3 Similarity measure

To compute similarity between Q_i and I_{DH}, similarity functions are required. In this work, color similarity (C_{ss}) is computed by using cosine distance (19). Similarity of texture (T_{ss}) and shape (S_{ss}) features is computed by using Euclidean distance (20).

$$C_{ss} = Sim(Q^c, IF_i^c, 'CD') \tag{16}$$
$$T_{ss} = Sim(Q^t, IF_i^t, 'ED') \tag{17}$$
$$S_{ss} = Sim(Q^s, IF_i^s, 'ED') \tag{18}$$

$$Cosine\ distance\ (CD) = \frac{\langle I^f \cdot Q^f \rangle}{\| I^f \| \| Q^f \|} \tag{19}$$

$$Euclidean\ distance\ (ED) = \sum_{i=1}^{|f|} \sqrt{\left(I^f - Q^f\right)^2} \tag{20}$$

4 Results and discussion

This section provides the implementation details of the proposed BiCBIR system and the datasets used for testing the performance. The proposed system has been developed and tested on Matlab 2017b software. The hardware configuration of the system used for this work as follow: Xenon(R) 2.60 GHz with 8GB of RAM and Windows 8.1 Pro.

4.1 Dataset

Proposed system is evaluated on two image datasets. **COREL,corel** has 1000 images containing hundred images of ten categories which include African people, Beaches, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Foods respectively. **GHIM-1k** is a subset of GHIM-10k dataset [4] which contains thousand images of ten categories (Fireworks, Building, Wall, Car, Bees, Mountains, Flowers, Trees, Fields and Beaches) with each category having hundred images.

Table 3 The CBIR variants are divided into three categories based on the number of layers used by the methods

Methods	Features used	Category description
C	Color	In this category, different methods, compute the similarity in single layer and on single feature space in C, T and S retrieval methods whereas $CTS1$ combines color, texture and shape features.
T	Texture	
S	Shape	
$CTS1$	Color + Texture + Shape	
TCCS	Layer1: Color + Texture and Layer2: Color + Shape	This category contains the image retrieval methods which compute the similarity in two layers. First layer computes the similarity of query image with all N images in the dataset while second layer computes similarity of only M images having similarity score more than the rest of the images where M is 20% of N.
CTTS	Layer1: Texture + Color and Layer2: Texture + Shape	
CSST	Layer1: Shape + Color and Layer2: Shape + Texture	
SCCT	Layer1: Color + Shape and Layer2: Color + Texture	
STTC	Layer1: Texture + Shape and Layer2: Texture + Color	
TSSC	Layer1: Shape + Texture and Layer2: Shape + Color	
CTS	Layer1: Color, Layer2: Texture and Layer3: Shape	Methods belonging to this category comprise of three layers. First layer compares all the N images with query image and returns M_1 images to second layer, second layer compares only M_1 images and returns M_2 images to third layer. Third layer compares M_2 images and finally returns F images to the end user, where $N \gg M_1 > M_2 > F$ (M_1 is 10% of N, M_2 is 20% of N and F is 20).
CST	Layer1: Color, Layer2: Shape and Layer3: Texture	
TCS	Layer1: Texture, Layer2: Color and Layer3: Shape	
TSC	Layer1: Texture, Layer2: Shape and Layer3: Color	
STC	Layer1: Shape, Layer2: Texture and Layer3: Color	
SCT	Layer1: Shape, Layer2: Color and Layer3: Texture	

4.2 Evaluation parameters

Performance of the proposed BiCBIR system is evaluated on the basis of precision (Pr), recall (Re) and f-score (Fs) @20. The evaluation parameters are given in (21), (22) and (23), respectively.

$$Pr = \frac{|I(Ret) \cap I(Rel)|}{|I(Ret)|} \tag{21}$$

$$Re = \frac{|I(Ret) \cap I(Rel)|}{|I(Rel)|} \tag{22}$$

$$Fs = \frac{2 \times Pr \times Re}{Pr + Re} \tag{23}$$

Here I(Ret) are retrieved images and I(Rel) are relevant images.

4.3 CBIR models

Experiments are performed on two datasets listed in Section 4.1 by using three CBIR models, first model consists of four variants, second model consists of six variants and third model consists of six variants. The variants of CBIR models are described in Table 3.

4.4 Experimental results

In this section experimental results and analysis are provided. Results are obtained on the basis of precision, recall and f-score. The experimental results are provided in two subsections, Sections 4.4.1 and 4.4.3. The former discusses results for sixteen variants of CBIR based on type of features, number of features, sequence of features and number of layers used for image retrieval; while in the latter, the best method is selected which is further compared with other state-of-the-art methods. The bar graphs are generated for each category of CBIR model, evaluation parameters and datasets. Results shown in tables include all sixteen variants of CBIR.

4.4.1 Variants of CBIR

The experiments are performed on sixteen variants of image retrieval system which are grouped into three categories; a) Single layer CBIR model, b) Bi-layer CBIR model and c) Tri)-layer CBIR model. These variants are tested on two datasets COREL [5] and GHIM [4].

Performance of single layer CBIR systems is demonstrated in Fig. 4a, b and c, respectively for COREL dataset. Figure 4d, e and f show the precision, recall and f-score for GHIM dataset. It has been observed from the Fig. 4a and d that in both COREL and GHIM dataset, in comparison to single feature, better results are produced when all three features are combined. Further, the results vary for different datasets when single feature is considered; for COREL dataset shape feature performs well while texture feature provide best retrieval results for GHIM dataset.

Image retrieval results of two layer CBIR systems depicted in Fig. 5a, b and c show the precision, recall and f-score, respectively for COREL dataset and Fig. 5d, e and f show the precision, recall and f-score for GHIM dataset. It has been observed from the Fig. 5a that TSSC feature sequence provide best and consistent results for all the categories of images for COREL dataset.

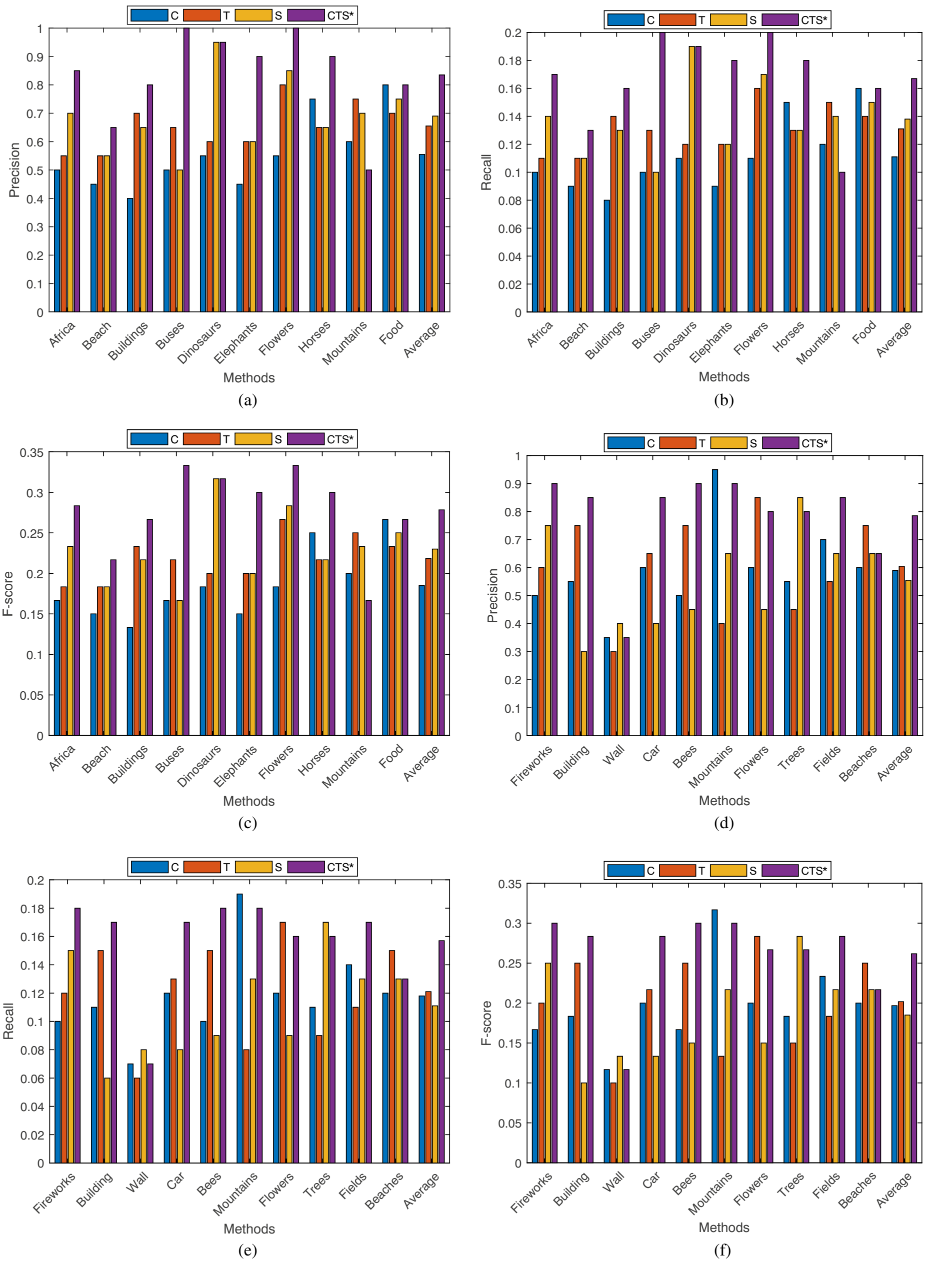

Fig. 4 Image retrieval results of single layer CBIR variants on (4**a**: precision, 4**b**: recall, 4**c**: f-score) COREL dataset and (4**d**: precision, 4**e**: recall, 4**f**: f-score) GHIM dataset

The retrieval results of CBIR systems under the three layer CBIR model are demonstrated in Fig. 6a, b and c for COREL dataset and Fig. 6d, e and f shows the performance on GHIM dataset. As observed from Fig. 6a and d, result are not good for all categories with any of

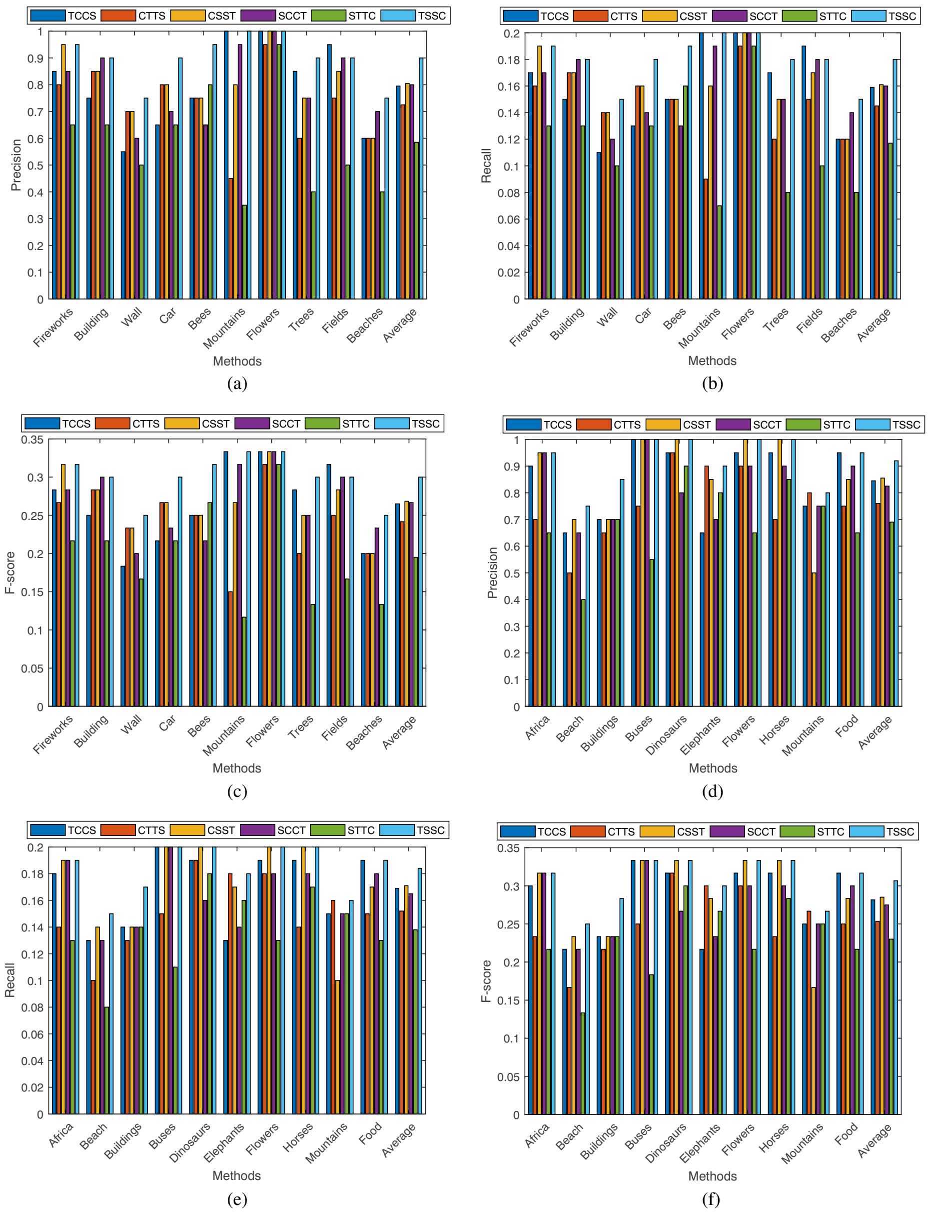

Fig. 5 Image retrieval results of CBIR variants belongs to two layer CBIR model (5**a**, 5**b** and 5**c** are the precision, recall and f-score, respectively) on COREL dataset and (5**d**, 5**e** and 5**f** are the precision, recall and f-score, respectively) GHIM dataset

the above mentioned method. Due to this inconsistency in retrieval performance, average retrieval for all the image categories is considered for performance evaluation. It has been observed from the Fig. 6a that TCS method has high retrieval rate for COREL dataset while CST has for GHIM dataset.

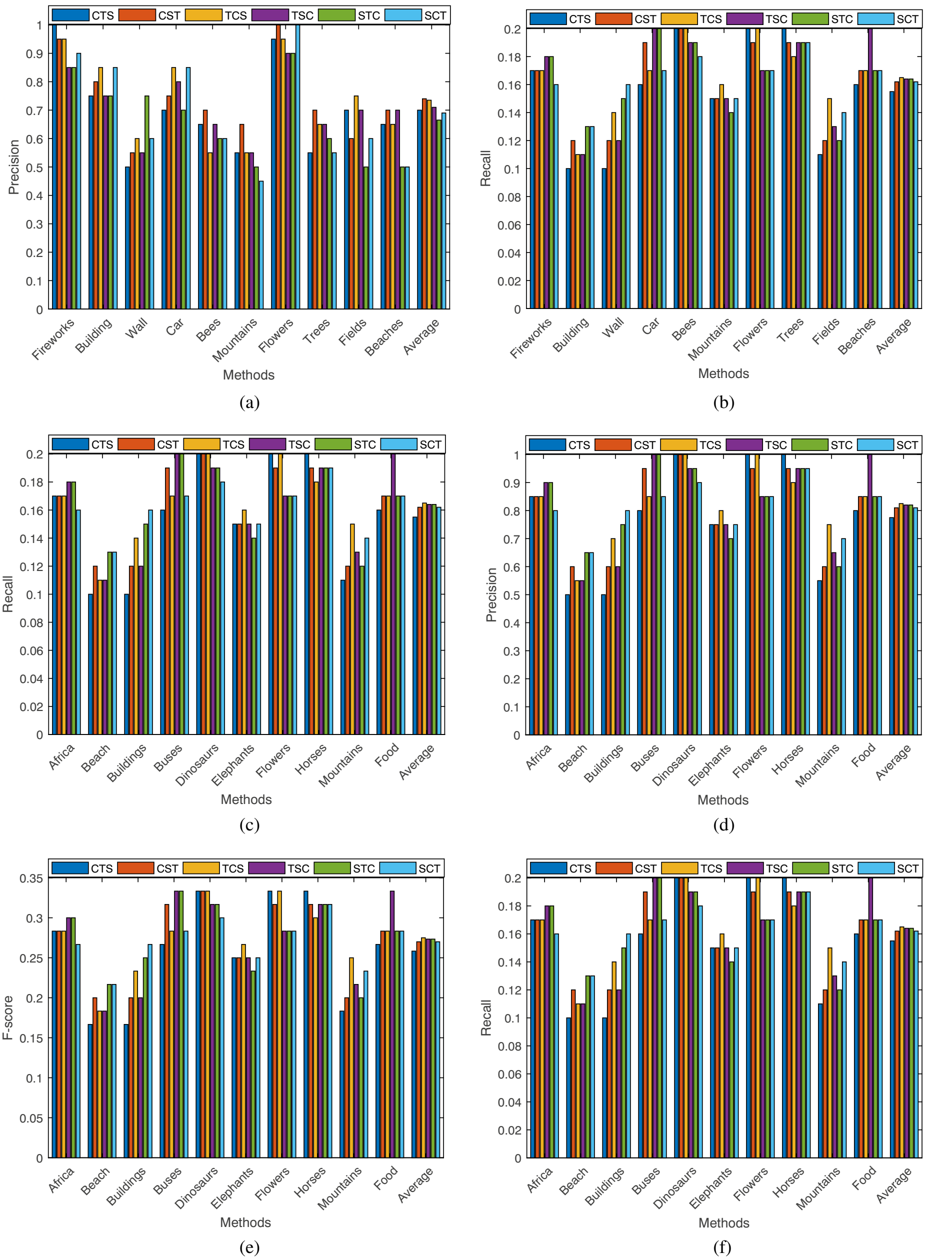

Fig. 6 Image retrieval results of CBIR variants in three layer CBIR model on COREL dataset 6**a**, 6**b** and 6**c** are the precision, recall and f-score, respectively) and GHIM dataset (6**d**, 6**e** and 6**f** are the precision, recall and f-score, respectively)

From the above discussion (Tables 4 and 5) it can be concluded that the sequence of features texture plus shape at first layer and shape plus color at second layer *i.e.* TSSC outperforms all the other sequences in both datasets.

Table 4 Retrieval performance of CBIR variants on COREL dataset in terms of precision, recall and f-score

Class	Evalua tion para meter	C	T	S	CTS1	TCCS	CTTS	CSST	SCCT	STTC	TSSC	CTS	CST	TCS	TSC	STC	SCT
Africa	Preci sion	0.500	0.550	0.700	0.850	0.900	0.700	0.950	0.950	0.650	0.950	0.850	0.850	0.850	0.900	0.900	0.800
	Recall	0.100	0.110	0.140	0.170	0.180	0.140	0.190	0.190	0.130	0.190	0.170	0.170	0.170	0.180	0.180	0.160
	Fscore	0.167	0.183	0.233	0.283	0.300	0.233	0.317	0.317	0.217	0.317	0.283	0.283	0.283	0.300	0.300	0.267
Beach	Preci sion	0.450	0.550	0.550	0.650	0.650	0.500	0.700	0.650	0.400	0.750	0.500	0.600	0.550	0.550	0.650	0.650
	Recall	0.090	0.110	0.110	0.130	0.130	0.100	0.140	0.130	0.080	0.150	0.100	0.120	0.110	0.110	0.130	0.130
	Fscore	0.150	0.183	0.183	0.217	0.217	0.167	0.233	0.217	0.133	0.250	0.167	0.200	0.183	0.183	0.217	0.217
Buil ding	Preci sion	0.400	0.700	0.650	0.800	0.700	0.650	0.700	0.700	0.700	0.850	0.500	0.600	0.700	0.600	0.750	0.800
	Recall	0.080	0.140	0.130	0.160	0.140	0.130	0.140	0.140	0.140	0.170	0.100	0.120	0.140	0.120	0.150	0.160
	Fscore	0.133	0.233	0.217	0.267	0.233	0.217	0.233	0.233	0.233	0.283	0.167	0.200	0.233	0.200	0.250	0.267
Bus	Preci sion	0.500	0.650	0.500	1.000	1.000	0.750	1.000	1.000	0.550	1.000	0.800	0.950	0.850	1.000	1.000	0.850
	Recall	0.100	0.130	0.100	0.200	0.200	0.150	0.200	0.200	0.110	0.200	0.160	0.190	0.170	0.200	0.200	0.170
	Fscore	0.167	0.217	0.167	0.333	0.333	0.250	0.333	0.333	0.183	0.333	0.267	0.317	0.283	0.333	0.333	0.283
Dino saur	Preci sion	0.550	0.600	0.950	0.950	0.950	0.950	1.000	0.800	0.900	1.000	1.000	1.000	1.000	0.950	0.950	0.900
	Recall	0.110	0.120	0.190	0.190	0.190	0.190	0.200	0.160	0.180	0.200	0.200	0.200	0.200	0.190	0.190	0.180
	Fscore	0.183	0.200	0.317	0.317	0.317	0.317	0.333	0.267	0.300	0.333	0.333	0.333	0.333	0.317	0.317	0.300
Ele phant	Preci sion	0.450	0.600	0.600	0.900	0.650	0.900	0.850	0.700	0.800	0.900	0.750	0.750	0.800	0.750	0.700	0.750
	Recall	0.090	0.120	0.120	0.180	0.130	0.180	0.170	0.140	0.160	0.180	0.150	0.150	0.160	0.150	0.140	0.150
	Fscore	0.150	0.200	0.200	0.300	0.217	0.300	0.283	0.233	0.267	0.300	0.250	0.250	0.267	0.250	0.233	0.250
Flo wer	Preci sion	0.550	0.800	0.850	1.000	0.950	0.900	1.000	0.900	0.650	1.000	1.000	0.950	1.000	0.850	0.850	0.850
	Recall	0.110	0.160	0.170	0.200	0.190	0.180	0.200	0.180	0.130	0.200	0.200	0.190	0.200	0.170	0.170	0.170
	Fscore	0.183	0.267	0.283	0.333	0.317	0.300	0.333	0.300	0.217	0.333	0.333	0.317	0.333	0.283	0.283	0.283
Horse	Preci sion	0.750	0.650	0.650	0.900	0.950	0.700	1.000	0.900	0.850	1.000	1.000	0.950	0.900	0.950	0.950	0.950

Table 4 (continued)

Class	Evaluation parameter	C	T	S	CTS1	TCCS	CTTS	CSST	SCCT	STTC	TSSC	CTS	CST	TCS	TSC	STC	SCT
	Recall	0.150	0.130	0.130	0.180	0.190	0.140	0.200	0.180	0.170	0.200	0.200	0.190	0.180	0.190	0.190	0.190
	Fscore	0.250	0.217	0.217	0.300	0.317	0.233	0.333	0.300	0.283	0.333	0.333	0.317	0.300	0.317	0.317	0.317
Mountain	Precision	0.600	0.750	0.700	0.500	0.750	0.800	0.500	0.750	0.750	0.800	0.550	0.600	0.750	0.650	0.600	0.700
	Recall	0.120	0.150	0.140	0.100	0.150	0.160	0.100	0.150	0.150	0.160	0.110	0.120	0.150	0.130	0.120	0.140
	Fscore	0.200	0.250	0.233	0.167	0.250	0.267	0.167	0.250	0.250	0.267	0.183	0.200	0.250	0.217	0.200	0.233
Food	Precision	0.800	0.700	0.750	0.800	0.950	0.750	0.850	0.900	0.650	0.950	0.800	0.850	0.850	1.000	0.850	0.850
	Recall	0.160	0.140	0.150	0.160	0.190	0.150	0.170	0.180	0.130	0.190	0.160	0.170	0.170	0.200	0.170	0.170
	Fscore	0.267	0.233	0.250	0.267	0.317	0.250	0.283	0.300	0.217	0.317	0.267	0.283	0.283	0.333	0.283	0.283
Average	Precision	0.555	0.655	0.690	0.835	0.845	0.760	0.855	0.825	0.690	0.920	0.775	0.810	0.825	0.820	0.820	0.810
	Recall	0.111	0.131	0.138	0.167	0.169	0.152	0.171	0.165	0.138	0.184	0.155	0.162	0.165	0.164	0.164	0.162
	Fscore	0.185	0.218	0.230	0.278	0.282	0.253	0.285	0.275	0.230	0.307	0.258	0.270	0.275	0.273	0.273	0.270

Table 5 Retrieval performance of CBIR variants on GHIM dataset in terms of precision, recall and f-score

Class	Evaluation	C	T	S	CTS1	TCCS	CTTS	CSST	SCCT	STTC	TSSC	CTS	CST	TCS	TSC	STC	SCT
Fireworks	Precision	0.500	0.600	0.750	0.900	0.850	0.800	0.950	0.850	0.650	0.950	1.000	0.950	0.950	0.850	0.850	0.900
	Recall	0.100	0.120	0.150	0.180	0.170	0.160	0.190	0.170	0.130	0.190	0.200	0.190	0.190	0.170	0.170	0.180
	Fscore	0.167	0.200	0.250	0.300	0.283	0.267	0.317	0.283	0.217	0.317	0.333	0.317	0.317	0.283	0.283	0.300
Building	Precision	0.550	0.750	0.300	0.850	0.750	0.850	0.850	0.900	0.650	0.900	0.750	0.800	0.850	0.750	0.750	0.850
	Recall	0.110	0.150	0.060	0.170	0.150	0.170	0.170	0.180	0.130	0.180	0.150	0.160	0.170	0.150	0.150	0.170
	Fscore	0.183	0.250	0.100	0.283	0.250	0.283	0.283	0.300	0.217	0.300	0.250	0.267	0.283	0.250	0.250	0.283
Wall	Precision	0.350	0.300	0.400	0.350	0.550	0.700	0.700	0.600	0.500	0.750	0.500	0.550	0.600	0.550	0.750	0.600
	Recall	0.070	0.060	0.080	0.070	0.110	0.140	0.140	0.120	0.100	0.150	0.100	0.110	0.120	0.110	0.150	0.120
	Fscore	0.117	0.100	0.133	0.117	0.183	0.233	0.233	0.200	0.167	0.250	0.167	0.183	0.200	0.183	0.250	0.200
Car	Precision	0.600	0.650	0.400	0.850	0.650	0.800	0.800	0.700	0.650	0.900	0.700	0.750	0.850	0.800	0.700	0.850
	Recall	0.120	0.130	0.080	0.170	0.130	0.160	0.160	0.140	0.130	0.180	0.140	0.150	0.170	0.160	0.140	0.170
	Fscore	0.200	0.217	0.133	0.283	0.217	0.267	0.267	0.233	0.217	0.300	0.233	0.250	0.283	0.267	0.233	0.283
Bees	Precision	0.500	0.750	0.450	0.900	0.750	0.750	0.750	0.650	0.800	0.950	0.650	0.700	0.550	0.650	0.600	0.600
	Recall	0.100	0.150	0.090	0.180	0.150	0.150	0.150	0.130	0.160	0.190	0.130	0.140	0.110	0.130	0.120	0.120
	Fscore	0.167	0.250	0.150	0.300	0.250	0.250	0.250	0.217	0.267	0.317	0.217	0.233	0.183	0.217	0.200	0.200
Mountains	Precision	0.950	0.400	0.650	0.900	1.000	0.450	0.800	0.950	0.350	1.000	0.550	0.650	0.550	0.550	0.500	0.450
	Recall	0.190	0.080	0.130	0.180	0.200	0.090	0.160	0.190	0.070	0.200	0.110	0.130	0.110	0.110	0.100	0.090
	Fscore	0.317	0.133	0.217	0.300	0.333	0.150	0.267	0.317	0.117	0.333	0.183	0.217	0.183	0.183	0.167	0.150
Flowers	Precision	0.600	0.850	0.450	0.800	1.000	0.950	1.000	1.000	0.950	1.000	0.950	1.000	0.950	0.900	0.900	1.000
	Recall	0.120	0.170	0.090	0.160	0.200	0.190	0.200	0.200	0.190	0.200	0.190	0.200	0.190	0.180	0.180	0.200
	Fscore	0.200	0.283	0.150	0.267	0.333	0.317	0.333	0.333	0.317	0.333	0.317	0.333	0.317	0.300	0.300	0.333
Trees	Precision	0.550	0.450	0.850	0.800	0.850	0.600	0.750	0.750	0.400	0.900	0.550	0.700	0.650	0.650	0.600	0.550
	Recall	0.110	0.090	0.170	0.160	0.170	0.120	0.150	0.150	0.080	0.180	0.110	0.140	0.130	0.130	0.120	0.110
	Fscore	0.183	0.150	0.283	0.267	0.283	0.200	0.250	0.250	0.133	0.300	0.183	0.233	0.217	0.217	0.200	0.183

Table 5 (continued)

Class	Evaluation	C	T	S	CTS1	TCCS	CTTS	CSST	SCCT	STTC	TSSC	CTS	CST	TCS	TSC	STC	SCT
Fields	Precision	0.700	0.550	0.650	0.850	0.950	0.750	0.850	0.900	0.500	0.900	0.700	0.600	0.750	0.700	0.500	0.600
	Recall	0.140	0.110	0.130	0.170	0.190	0.150	0.170	0.180	0.100	0.180	0.140	0.120	0.150	0.140	0.100	0.120
	Fscore	0.233	0.183	0.217	0.283	0.317	0.250	0.283	0.300	0.167	0.300	0.233	0.200	0.250	0.233	0.167	0.200
Beaches	Precision	0.600	0.750	0.650	0.650	0.600	0.600	0.600	0.700	0.400	0.750	0.650	0.700	0.650	0.700	0.500	0.500
	Recall	0.120	0.150	0.130	0.130	0.120	0.120	0.120	0.140	0.080	0.150	0.130	0.140	0.130	0.140	0.100	0.100
	Fscore	0.200	0.250	0.217	0.217	0.200	0.200	0.200	0.233	0.133	0.250	0.217	0.233	0.217	0.233	0.167	0.167
Average	Precision	0.590	0.605	0.555	0.785	0.795	0.725	0.805	0.800	0.585	0.900	0.700	0.740	0.735	0.710	0.665	0.690
	Recall	0.118	0.121	0.111	0.157	0.159	0.145	0.161	0.160	0.117	0.180	0.140	0.148	0.147	0.142	0.133	0.138
	Fscore	0.197	0.202	0.185	0.262	0.265	0.242	0.268	0.267	0.195	0.300	0.233	0.247	0.245	0.237	0.222	0.230

4.4.2 Analysis based on number of processing steps

This section provides analysis of the single layer, bi-layer (proposed) and tri-layer CBIR systems on the basis of number of processing steps (PS). In next step, the number of steps required for processing are calculated and the best method among each category of CBIR variants is selected. For single layer, bi-layer and tri-layer, **CST1**, **TSSC** and **TCS** CBIR approaches are chosen, respectively.

Let F_1, F_2 and F_3 be the color, texture and shape feature vectors, respectively. The size of feature vectors F_1, F_2 and F_3 are considered as 81, 60 and 21, respectively.

$$F_1 \approxeq \frac{4}{3} \times F_2 \tag{24}$$

$$F_1 \approxeq 4 \times F_3 \tag{25}$$

$$F_2 \approxeq 3 \times F_3 \tag{26}$$

From the length of feature vectors, (24), (25) and (26) are derived. For single layer, bi-layer and tri-layer approach, consider N as the number of images in the dataset.

Single layer approach (CST1): In this approach, number of processing steps (PS) required are:

$$PS = N \times [|F_1| + |F_2| + |F_3|] \tag{27}$$

By substituting F_1 and F_2 with F_3 from (25) and (26), we get $PS \approxeq N \times [4 \times |F_3| + 3 \times |F_3| + |F_3|]$

$$PS \approxeq 8 \times N \times F_3 \tag{28}$$

Bi-layer approach (TSSC): Total number of steps required are:

$$TPS = N \times |F_2 + F_3| + \frac{N}{5} \times |F_1 + F_3| \tag{29}$$

By substituting F_1 and F_2 with F_3 from (25) and (26), we get (32)

$$TPS \approxeq N \times |3 \times F_3 + F_3| + \frac{N}{5} \times |4 \times F_3 + F_3| \tag{30}$$

$$TPS \approxeq N \times |4 \times F_3| + \frac{N}{5} \times |5 \times F_3| \tag{31}$$

$$TPS \approxeq 5 \times N \times |F_3| \tag{32}$$

with $5 \times N \times F_3$ PS, which is only 0.625 times of approach $CTS1$.

Tri-layer approach (TSC): Processing steps required in this approach are:

$$TPS = N \times |F_2| + \frac{N}{10} \times |F_1| + \frac{N}{20} \times F_3 \tag{33}$$

By substituting F_1 and F_2 with F_3 from (25) and (26), we get (34)

$$TPS \approxeq N \times 3 \times |F_3| + \frac{N}{10} \times 4 \times |F_3| + \frac{N}{20} \times |F_3| \tag{34}$$

$$TPS \approxeq \frac{69 \times N \times |F_3|}{20} \tag{35}$$

with $3.45 \times N \times F_3$ PS, which takes less than half of PS as compared to approach $CTS1$ and slightly lesser than the proposed approach. The proposed approach takes slightly larger processing steps compared to TCS approach but this extra computational cost provides 10% (Table 4) higher average precision for WANG dataset and 17% (Table 5) for GHIM dataset.

Fig. 7 Proposed CBIR system is compared with five related CBIR systems, Elalami et al. [6], Zeng et al. [48], Guo et al. [11], Lande et al. [13] and Pradhan et al. [26] (7**a** demonstrate the precision, 7**a** shows the recall and **7a** depict the f-score) on COREL dataset

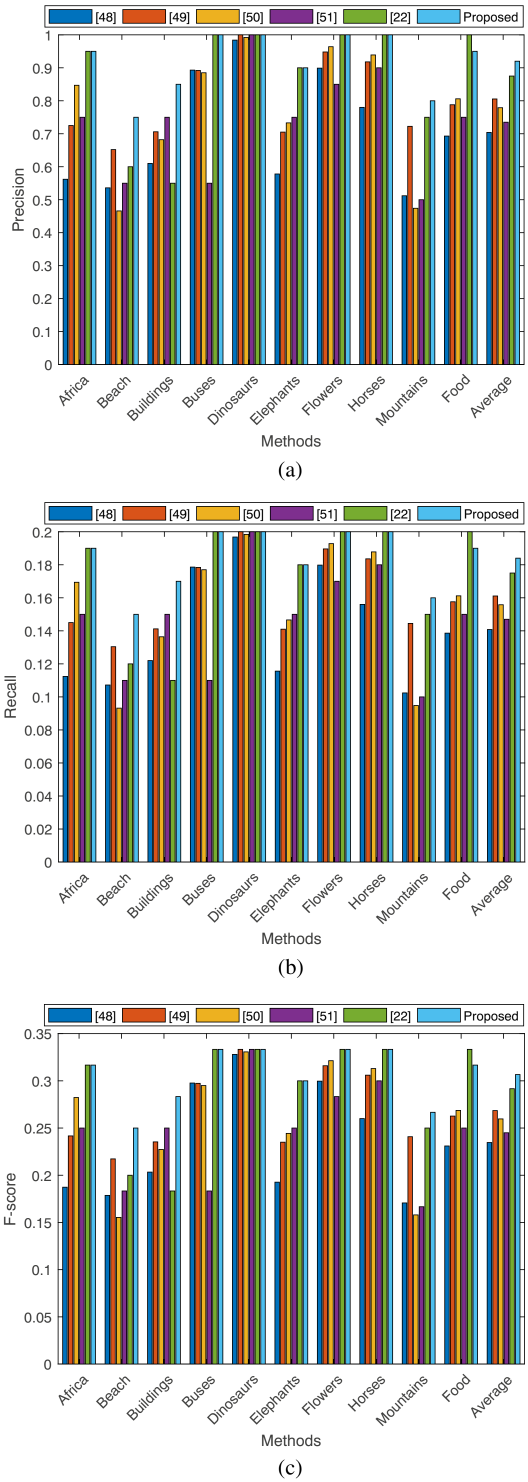

Table 6 Comparison of proposed BiCBIR system with five other CBIR systems on COREL dataset in term of precision, recall and f-score

Class	Evaluation	Elalami et al. [6]	Zeng et al. [48]	Guo et al. [11]	Lande et al. [13]	Pradhan et al. [26]	Proposed
Africa	Precision	0.562	0.725	0.847	0.750	0.950	0.950
	Recall	0.112	0.145	0.169	0.150	0.190	0.190
	Fscore	0.187	0.242	0.282	0.250	0.317	0.317
Beach	Precision	0.536	0.652	0.466	0.550	0.600	0.750
	Recall	0.107	0.130	0.093	0.110	0.120	0.150
	Fscore	0.179	0.217	0.155	0.183	0.200	0.250
Building	Precision	0.610	0.706	0.682	0.750	0.550	0.850
	Recall	0.122	0.141	0.136	0.150	0.110	0.170
	Fscore	0.203	0.235	0.227	0.250	0.183	0.283
Bus	Precision	0.893	0.892	0.885	0.550	1.000	1.000
	Recall	0.179	0.178	0.177	0.110	0.200	0.200
	Fscore	0.298	0.297	0.295	0.183	0.333	0.333
Dinosaur	Precision	0.984	1.000	0.992	1.000	1.000	1.000
	Recall	0.197	0.200	0.198	0.200	0.200	0.200
	Fscore	0.328	0.333	0.331	0.333	0.333	0.333
Elephant	Precision	0.578	0.705	0.733	0.750	0.900	0.900
	Recall	0.116	0.141	0.147	0.150	0.180	0.180
	Fscore	0.193	0.235	0.244	0.250	0.300	0.300
Flower	Precision	0.899	0.948	0.964	0.850	1.000	1.000
	Recall	0.180	0.190	0.193	0.170	0.200	0.200
	Fscore	0.300	0.316	0.321	0.283	0.333	0.333
Horse	Precision	0.780	0.918	0.939	0.900	1.000	1.000
	Recall	0.156	0.184	0.188	0.180	0.200	0.200
	Fscore	0.260	0.306	0.313	0.300	0.333	0.333

Table 6 (continued)

Class	Evaluation	Elalami et al. [6]	Zeng et al. [48]	Guo et al. [11]	Lande et al. [13]	Pradhan et al. [26]	Proposed
Mountain	Precision	0.512	0.723	0.474	0.500	0.750	0.800
	Recall	0.102	0.145	0.095	0.100	0.150	0.160
	Fscore	0.171	0.241	0.158	0.167	0.250	0.267
Food	Precision	0.693	0.788	0.806	0.750	1.000	0.950
	Recall	0.139	0.158	0.161	0.150	0.200	0.190
	Fscore	0.231	0.263	0.269	0.250	0.333	0.317
Average	Precision	0.705	0.806	0.779	0.735	0.875	0.920
	Recall	0.141	0.161	0.156	0.147	0.175	0.184
	Fscore	0.235	0.269	0.260	0.245	0.292	0.307

On the basis of precision rate and processing steps $TSSC$ is selected as proposed BiCBIR method. Further it is compared with other state-of-the-art CBIR methods.

4.4.3 Comparison of proposed CBIR (BiCBIR) system with other CBIR systems

The proposed BiCBIR approach has been compared with some of the existing CBIR systems discussed in this section. Elalami et al. [6] proposed the integration of the color coherence vector and wavelet features to enhance the retrieval performance. Zeng et al. [48] compute the similarity between two spatiograms through a similarity measure function based on Jensen–Shannon Divergence. Guo and Prasetyo [11] proposed two image features, color co-occurrence and bit pattern features. To achieve fast search, images are indexed directly from the ODBTC encoded data streams without performing the decoding process. CBIR system proposed by Lande et al. [13] uses color, texture, and shape features of an image. The extracted features are combined to improve the efficacy for matching and retrieval purpose. In [26] Pradhan et al. proposed a hierarchical CBIR system which consists of three layers and uses three image features such as color, texture and shape. This hierarchical CBIR model reduces the search space at each layer by removing the non-relevant images to achieve high retrieval speed.

It is evident from Fig. 7a that in nine categories of images for precision of the proposed system is better or equal in COREL dataset. Recall and f-score are shown in Fig. 7b and c, respectively. Results are provided in Table 6 to quantify the efficiency of the proposed CBIR system, which shows average retrieval for all image categories as better than other five CBIR systems. In Table 6, precision of proposed BiCBIR system have been depicted along with other state-of-the-art methods.

4.5 Discussion

Image retrieval approaches especially CBIR approaches extract image features from the given images. The most commonly used image features (also known as primitive image features) are color, texture and shape; used to compare and retrieve the images most similar to query image. In majority of the proposed CBIR systems, query image results are retrieved by searching all the image feature types in the entire dataset, leading to high computational time. An alternate approach is to compare query image with subset of feature space in multiple layers. But one of the major shortcomings of layered approach is that images pruned based on feature considered in first layer are not considered in the next layer although they may be similar to the query image. The proposed scheme named BiCBIR is a hybrid of both the above mentioned approaches, where retrieval of similar images takes place in two stages and two features are considered in both the layers. Among the three features considered for similarity search, one feature remains common in both layers, leading to less search time and improved retrieval accuracy.

5 Conclusion

In this work, a CBIR system (BiCBIR) has been proposed which uses color, texture and shape features to represent query and dataset images. The retrieval module of the proposed system consists of two layers; in the first layer, it compare texture and shape features and most relevant images are passed to the next layer. In the second layer, color and shape features are matched and most similar images are returned to the user in response to the query image. Experimental results indicate that the proposed system is accurate and faster. Identification of the best feature sequence is done experimentally and results show that retrieval rate is high when texture and shape features are considered at first layer and shape and color features at second layer. Proposed system is a layered based image retrieval system and all images with varying type and size can be incorporated. In future work, proposed system will be extended with convolution neural networks based image features which can further improve the performance of the BiCBIR.

References

1. Ahmed KT, Ummesafi S, Iqbal A (2019) Content based image retrieval using image features information fusion. Inf Fus 51:76–99
2. Chang X, Ma Z, Yi Y, Zeng Z, Hauptmann AG (2016) Bi-level semantic representation analysis for multimedia event detection. IEEE Trans Cybern 47(5):1180–1197
3. Cheng Z, Chang X, Zhu L, Kanjirathinkal RC, Kankanhalli M (2019) Mmalfm: Explainable recommendation by leveraging reviews and images. ACM Trans Inf Syst (TOIS) 37(2):16
4. Cifar-10 database. http://www.cs.toronto.edu/~kriz/cifar.html. Last accessed: 2018-4-10
5. Corel database. http://wang.ist.psu.edu/docs/home.shtml. Last accessed: 2018-4-10
6. Esmel ElAlami M (2011) A novel image retrieval model based on the most relevant features. Knowl-Based Syst 24(1):23–32
7. Fadaei S, Amirfattahi R, Ahmadzadeh MR (2016) New content-based image retrieval system based on optimised integration of dcd, wavelet and curvelet features, vol 11
8. Gabor D (1946) Theory of communication. part 1: The analysis of information. J Inst Electr Eng-Part III: Radio Commun Eng 93(26):429–441
9. Gao Z, Wang DY, Wan SH, Zhang H, Wang YL (2019) Cognitive-inspired class-statistic matching with triple-constrain for camera free 3d object retrieval. Futur Gener Comput Syst 94:641–653
10. Garcia N, Vogiatzis G (2019) Learning non-metric visual similarity for image retrieval. Image Vis Comput 82:18–25
11. Guo J-M, Prasetyo H (2015) Content-based image retrieval using features extracted from halftoning-based block truncation coding. IEEE Trans Image Process 24(3):1010–1024
12. Jian M, Yin Y, Dong J, Lam K-M (2018) Content-based image retrieval via a hierarchical-local-feature extraction scheme. Multimedia Tools and Applications 77(21):29099–29117
13. Lande MV, Bhanodiya P, Jain P (2014) An effective content-based image retrieval using color, texture and shape feature. In: Intelligent computing, networking, and informatics. Springer, pp 1163–1170
14. Liu G-H, Yang J-Y (2013) Content-based image retrieval using color difference histogram. Pattern Recogn 46(1):188–198
15. Liu Z, Li H, Zhang L, Zhou W, Qi T (2014) Cross-indexing of binary sift codes for large-scale image search. IEEE Trans Image Process 23(5):2047–2057
16. Ma Z, Chang X, Xu Z, Sebe N, Hauptmann AG (2017) Joint attributes and event analysis for multimedia event detection. IEEE Trans Neural Netw Learn Syst 29(7):2921–2930
17. Mahmood T, Mehmood Z, Shah M, Khan Z (2018) An efficient forensic technique for exposing region duplication forgery in digital images. Appl Intell 48(7):1791–1801
18. Mansoori N, Nejati M, Razzaghi P, Samavi S (2013) Image retrieval by bag of visual words and color information. In: The 21st Iranian Conference on Electrical Engineering (ICEE), Mashhad
19. Marin-Castro HM, Hernandez-Resendiz JD, Escalante-Balderas HJ, Pellegrin L, Tello-Leal E (2019) Chained ensemble classifier for image annotation. Multimedia Tools and Applications 78(18):26263–26285
20. Mehmood Z, Mahmood T, Javid MA (2018) Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine. Appl Intell 48(1):166–181
21. Mezzoudj S, Seghir R, Saadna Y et al (2019) A parallel content-based image retrieval system using spark and tachyon frameworks. Journal of King Saud University-Computer and Information Sciences
22. Ouni A, Urruty T, Visani M (2018) A robust cbir framework in between bags of visual words and phrases models for specific image datasets. Multimedia Tools and Applications 77(20):26173–26189
23. Pandey S, Khanna P (2016) Content-based image retrieval embedded with agglomerative clustering built on information loss. Comput Electr Eng 54:506–521

24. Pavithra LK, Sree Sharmila T (2017) An efficient framework for image retrieval using color, texture and edge features. Computers & Electrical Engineering
25. Phadikar BS, Phadikar A, Maity GK (2018) Content-based image retrieval in dct compressed domain with mpeg-7 edge descriptor and genetic algorithm. Pattern Anal Appl 21(2):469–489
26. Pradhan J, Kumar S, Pal AK, Banka H (2018) A hierarchical cbir framework using adaptive tetrolet transform and novel histograms from color and shape features. Digital Signal Processing
27. Ryszard S, Andrysiak CT, Choraś M (2007) Integrated color, texture and shape information for content-based image retrieval. Pattern Anal Appl 10(4):333–343
28. Shao J, Zhao Z, Su F (2019) Two-stage deep learning for supervised cross-modal retrieval. Multimedia Tools and Applications 78(12):16615–16631
29. Shrivastava N, Tyagi V (2014) Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching. Inf Sci 259:212–224
30. Shrivastava N, Tyagi V (2015) An efficient technique for retrieval of color images in large databases. Comput Electr Eng 46:314–327
31. Sotoodeh M, Moosavi MR, Boostani R (2019) A novel adaptive lbp-based descriptor for color image retrieval. Expert Systems with Applications
32. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vis 7(1):11–32
33. Talib A, Mahmuddin M, Husni H, George LE (2013) A weighted dominant color descriptor for content-based image retrieval. J Vis Commun Image Represent 24(3):345–360
34. Teague MR (1980) Image analysis via the general theory of moments. JOSA 70(8):920–930
35. Tong L, Tong R, Chen L (2019) Efficient retrieval algorithm for multimedia image information. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-019-07886-6
36. Tou JY, Tay YH, Lau PY (2007) Gabor filters and grey-level co-occurrence matrices in texture classification. In: MMU International symposium on information and communications technologies, pp 197–202
37. Tsai C-F (2012) Bag-of-words representation in image annotation: A review. ISRN Artificial Intelligence
38. Wang D, Ge S, Tan X (2019) Bayesian denoising hashing for robust image retrieval. Pattern Recogn 86:134–142
39. Wang X, Lee F, Chen Q (2019) Similarity-preserving hashing based on deep neural networks for large-scale image retrieval. J Vis Commun Image Represent 61:260–271
40. Wang X-Y, Li Y-W, Yang H-Y, Chen J-W (2014) An image retrieval scheme with relevance feedback using feature reconstruction and svm reclassification. Neurocomputing 127:214–230
41. Wang X-Y, Yu Y-J, Yang H-Y (2011) An effective image retrieval scheme using color, texture and shape features. Comput Stand Interfaces 33(1):59–68
42. Wang YL, Wang DY (2010) Clustering study of fabric deformation comfort using bi-swarm pso algorithm. J Text Res 31(4):60–64
43. Yan L, Lu H, Wang C, Ye Z, Chen H, Ling H (2019) Deep linear discriminant analysis hashing for image retrieval. Multimed Tools Appl 78(11):15101–15119
44. Yildizer E, Balci AM, Hassan M, Alhajj R (2012) Efficient content-based image retrieval using multiple support vector machines ensemble. Expert Syst Appl 39(3):2385–2396
45. Yousuf M, Mehmood Z, Habib HA, Mahmood T, Saba T, Rehman A, Rashid M (2018) A novel technique based on visual words fusion analysis of sparse features for effective content-based image retrieval. Mathematical Problems in Engineering
46. Yu Z, Wong H-S, You J, Han G (2012) Visual query processing for efficient image retrieval using a som-based filter-refinement scheme. Inf Sci 203:83–101
47. Yue J, Li Z, Lu L, Fu Z (2011) Content-based image retrieval using color and texture fused features. Math Comput Model 54(3-4):1121–1127
48. Zeng S, Huang R, Wang H, Kang Z (2016) Image retrieval using spatiograms of colors quantized by gaussian mixture models. Neurocomputing 171:673–684
49. Zhou W, Li H, Tian Q (2017) Recent advance in content-based image retrieval: A literature survey. arXiv:1706.06064
50. Zhou J, Liu X, Liu W, Gan J (2019) Image retrieval based on effective feature extraction and diffusion process. Multimedia Tools and Applications 78(5):6163–6190
51. Zhu S, Zou L, Fang B (2014) Content based image retrieval via a transductive model. J Intell Inf Syst 42(1):95–109

Sachendra Singh received M.E. Degree from Thapar Institute of Engineering and Technology, Patiala, India, in 2014. He is working as Research Scholar with Computer Science Department at Thapar Institute of Engineering and Technology, Punjab, India since July 2015. His research interest includes information retrieval, probabilistic data structures and similarity search.

Shalini Batra received the Ph.D. Degree in computer science and engineering from Thapar University, Patiala, India, in 2012. She is currently working as an Associate Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India. She has guided many research scholars leading to Ph.D. and M.E./M.Tech. She has authored more than 60 research papers published in various conferences and journals. Her research interest includes machine learning, web semantics, big data analytics and vehicular ad-hoc networks.