Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2020, Article ID 8851931, 20 pages
https://doi.org/10.1155/2020/8851931

Review Article

Image Retrieval Using Low Level and Local Features Contents: A Comprehensive Review

Jaya H. Dewan [1] **and Sudeep D. Thepade** [2]

[1]*Information Technology Department, Pimpri Chinchwad College of Engineering, Pune 411044, India*
[2]*Computer Engineering Department, Pimpri Chinchwad College of Engineering, Pune 411044, India*

Correspondence should be addressed to Sudeep D. Thepade; sudeepthepade@gmail.com

Received 24 April 2020; Revised 18 August 2020; Accepted 28 September 2020; Published 23 October 2020

Academic Editor: Cheng-Jian Lin

Billions of multimedia data files are getting created and shared on the web, mainly social media websites. The explosive increase in multimedia data, especially images and videos, has created an issue of searching and retrieving the relevant data from the archive collection. In the last few decades, the complexity of the image data has increased exponentially. Text-based image retrieval techniques do not meet the needs of the users due to the difference between image contents and text annotations associated with an image. Various methods have been proposed in recent years to tackle the problem of the semantic gap and retrieve images similar to the query specified by the user. Image retrieval based on image contents has attracted many researchers as it uses the visual content of the image such as color, texture, and shape feature. The low-level image features represent the image contents as feature vectors. The query image feature vector is compared with the dataset images feature vectors to retrieve similar images. The main aim of this article is to appraise the various image retrieval methods based on feature extraction, description, and matching content that has been presented in the last 10–15 years based on low-level feature contents and local features and proposes a promising future research direction for researchers.

1. Introduction

Humans have been using images for communication since pre-Roman times. Ancestors living in caves used to paint and carve pictures and maps on walls for communication. In the last two decades, exponential advances are visible in digital image processing technologies, network facilities, data repository technologies, smartphones, and cameras. This has resulted in videos and multimedia data being generated, uploaded on the Internet, and shared through social media websites, leading to an explosion in the amount and complexity of digital data being generated, stored, transmitted, analyzed, and accessed [1]. Access to a desired image from the repository involves searching for images portraying specific types of objects or scenes, identifying a particular mood, or simply searching the exact pattern or texture. The process of finding the desired image in a large and diverse collection is becoming a vital issue. The challenges in the field of image retrieval are becoming widely recognized, and the search for a solution is turning into a sought-after area for research.

The traditional way of an annotated image using text images are described using one or more keywords. It lacks the automatic and useful description of the image [2]. As compared to text retrieval, content-based image retrieval (CBIR) has been widely used in recent decades. Image retrieval using the content is considered one of the most successful and efficient ways of accessing visual data. This method is based on the image content such as shape, color, and texture instead of the annotated text.

The fundamental difficulties in image retrieval are the intention gap and the semantic gap. The problem to accurately convey the expected visual content using a query at hands, such as a sample image or a sketch map, is called the intention gap. The difficulty in depicting high-level semantic concepts using the low-level visual feature is called a

semantic gap [3]. Extensive efforts have been made by researchers to reduce these gaps.

There are three steps in image retrieval using contents: feature extraction, feature description, and image similarity measurement (Figure 1). An image is converted to some form of feature space for ease of comparison. The feature must be represented in the descriptive and discriminative form to differentiate related and unrelated images. The features extracted should be unaffected by various anomalies, such as differences in illumination, resizing, rotation, and translation changes.

For image retrieval using contents, a query is to be formed, which the user wants to search in the dataset. The query can be represented by giving an image as an input that can work as an example or reference. The text can be specified to search the dataset containing the object or a scene image similar to the text specified. The query can be given in the form of a sketch or clipart, which can work as a source to search the related images in the dataset, for example, a sketch of a human face, boat, or ball. A query can also be formed by specifying the color layout or concept present in the image. Here, the image retrieval system, which uses query specified with an example image, is in focus.

The article comprises the following sections: Section 2 discusses the various techniques based on color features. Section 3 describes the various techniques based on texture features. Sections 4 and 5 focus on shape features and local feature extraction techniques, respectively. Section 6 reviews the various feature fusion-based image retrieval techniques. Sections 7, 8, and 9 describe the various commonly available datasets, similarity measures, and performance measurement criteria used to evaluate the retrieval techniques, respectively. Section 10 presents the conclusion and future directions in image retrieval methods based on image contents.

2. Color Features Used in Image Retrieval

Color features are steady and robust as compared to other features. Most of the color features are invariant to scale, translation, and rotation changes. Various methods have been experimented and suggested in literature based on color features such as "color averaging," "color histogram," "color coherence," and "block truncation coding (BTC)" and its variants such as "Thepades Sorted Block Truncation Coding" (TSBTC)." The techniques developed based on color histograms have high effectiveness, simplicity, and low storage requirement.

Color moments are the low-level image features that can be used to measure the similarity between two images. The central color moments are standard deviation, mean, and skewness. These color moments give the distribution of colors in an image. Each image has green (G), red (R), and blue (B) color planes. Therefore, there are nine color moment features. The color moments can be defined as in the following equations.

$$\text{Mean}_i = \frac{1}{m * n} \sum_{j=1}^{m} \sum_{k=1}^{n} I(j,k,i), \tag{1}$$

$$\text{Standard deviation}_i = \sqrt{\frac{1}{m * n} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(I(j,k,i) - \text{mean}_i\right)^2}, \tag{2}$$

$$\text{Skewness}_i = \sqrt[3]{\frac{1}{m * n} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(I(j,k,i) - \text{mean}_i\right)^3}, \tag{3}$$

where i indicates the R, G, and B planes, and I (j, k, i) indicates the pixel intensities of the corresponding plane with size $m * n$.

The color histogram of the image is formed for R, G, and B planes. The histogram gives the probability distribution of the color intensities present in an image. The global histogram is calculated by considering the complete image as a whole. A local color histogram is calculated by dividing the image into parts, and then, the histogram of each part of the image is calculated. Color histograms are easy to compute and are less sensitive to small changes in viewpoints. Color histograms cannot provide spatial information and are sensitive to changes in illumination. There are various extensions of color histogram techniques such as the fuzzy histogram [4] and MPEG-7 dominant color descriptor [5]. In [5], the RGB color image is converted to HSV color space and is quantized to 72 levels for reducing the feature vector space (hue: 8 levels, S: 3 levels, and V: 3 levels). The histogram of the quantized image is used as a feature vector. Histogram intersection is used to find the images similar to the query image.

The color coherence vector technique [6] classifies each pixel as coherent if it is part of a large group of pixels having the same color; else, it is considered an incoherent pixel. The pixel group regions are created by connected components formed by checking the neighborhood pixel colors. If the connected component pixel count is greater than the threshold, it is considered a coherent region. The query images are compared with dataset images using the number of coherent pixels and incoherent pixels of a specific color.

The color correlogram technique represents the local spatial correlation of colors with the global distribution of these features [7]. Color correlogram represents the image as a table of color pair (p, q) where r entry in (p, q) cell indicates the probability of pixels with color q at a distance r from a pixel of color p. The color autocorrelogram technique represents the spatial correlation between the same color intensities.

In image retrieval using block truncation coding (BTC) [8, 9], the mean value is calculated for each plane of an image. The upper average is derived using pixels having a value above the mean. A lower average is derived using the

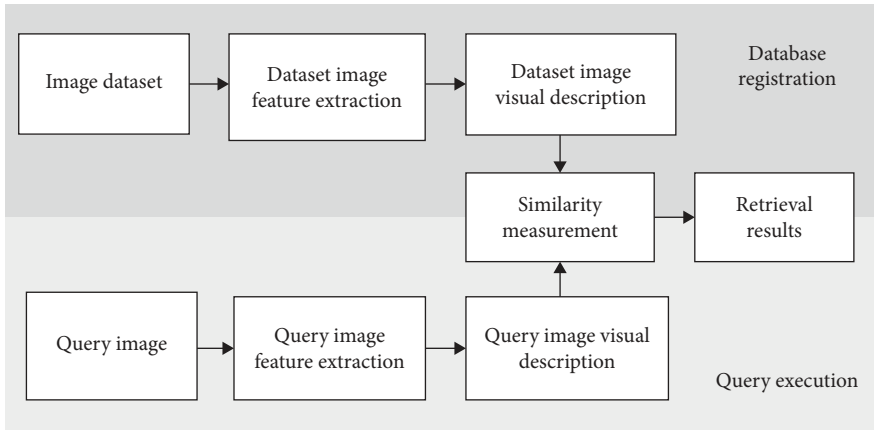

Figure 1: Block diagram of the image retrieval system based on contents.

pixels value having less than the mean. The pixels having a value less than the mean are assigned a value of lower average. If the pixel value is above the mean, then the pixel is assigned a value of upper average. This process can be applied iteratively by dividing the image into two blocks: the first block containing pixels having a value less than the mean and the second block containing pixels with a value greater than the mean. The upper average and lower average of the B, G, and R planes represent features of an image. A similar process is applied to query image, and matching images are retrieved from the dataset. There are various extensions of BTC such as "Dot Diffused BTC (DDBTC)" [10], "error diffused BTC" [11], "Optimized Dot Diffusion based BTC" [12], "halftoning based BTC" [13], and "BTC based on ant colony optimization" [14].

In image retrieval using TSBTC [15–17], the pixel values of an image are sorted in the ascending order, and the median value is calculated. Then, the lower mean is calculated using the pixel values below the median, and the upper mean is computed using pixel values above the median. This process is repeated iteratively on each block. Blocks are created by dividing the image into two parts with pixels below the median as one block and a pixel having a value above the median as another block. The upper and lower means of each color plane represent the features of the image which are used for image matching.

In [18], color planes of an image are binarized using the Niblack threshold selection method. These thresholds are used for calculating the upper and lower means of all the planes. These means and standard deviation for each plane are stored as the image feature vector. The image is compared using city block distance to identify the class of the image. The class of the image is only used for retrieving the relevant images. This method works well only if the query is directed to the correct class.

In [19], the RGB color space image is transformed to nonuniform HSV color space, and 72 color features are extracted from it through quantization. The color histogram is calculated from these features to find the dominant color features. The query image and dataset image similarity are measured using the dominance granule structure similarity method.

Table 1 shows a summary of various colors-based feature techniques. Color-based techniques are illumination variant but invariant to rotation and translation. The computational cost of color moment-based techniques is less, but accuracy is very low. The computational cost of histogram-based techniques is high, but accuracy is also high. The computational cost of BTC-based techniques is low, and accuracy is better.

3. Texture Features Used in Image Retrieval

The texture is another significant feature in retrieval techniques based on image contents. Image texture represents the variation in the local illumination in a small region. It represents the spatial layout of the gray intensities of pixels in a region. If the change of brightness is high in a small region, the image is called a coarse-textured image; else, it is called as a fine-textured image. The texture-based algorithms can be classified into two categories: statistical methods and structural methods. Structural methods identify the basic structure and their location in the image. These methods are useful in images containing textures that are very regular and works for images with human-made objects that have regular patterns. Statistical methods are simple and widely used methods that employ quantitative measurements of intensity arrangements in a region. Examples of such methods are the gray level histogram, edge histogram, "local binary pattern (LBP)" [20], "local ternary patterns (LTP)" [21], "Local Tetra Patterns (LTrP)" [22], "gray level co-occurrence matrix (GLCM)" [23], "wavelet coefficients" [24], "ridgelets and curvelets" [25], "Tamura features" [26], and "Gabor wavelet filter" [27].

LBP is a thresholding-based technique in which the center pixel is compared with its neighborhood pixels in the radius r. If the intensity value of the neighborhood pixels is larger than the center pixel, the code bit 0 is assigned to it; else, code bit 1 is assigned [20]. The binary codeword is generated for the center pixel by concatenating these neighborhood codewords. This codeword is then converted into a decimal number. The histogram of the decimal codewords is calculated for the image and can be used as the feature vector. The total number of bins in the histogram is

Table 1: Summary of color feature-based image retrieval techniques.

Year	Method	Similarity measure	Dataset	Performance measure (%)
2015	Dot-diffused BTC [10]	Modified Canberra	Corel-1000	Accuracy: 77.16
			Brodatz-1856	Accuracy: 81.19
			VisTex-640	Accuracy: 92.09
			STex	Accuracy: 44.79
			ALOT	Accuracy: 48.64
			OutexTC00013	Accuracy: 66.82
			KTHTIPS	Accuracy: 64.81
2016	Feature vector generation using Niblack binarizaion, classification using artificial neural network [18]	City block distance	Wang	Precision: 83.8 Recall: 83.7
			OT scene	Precision: 66.5 Recall: 66.3
2018	Color histogram using quantized HSV color space [19]	Improved dominance granule structure similarity method	COIL-20	Precision: 48.18 Recall: 83.87
			Corel-1000	Precision: 68.3 Recall: 37.9
2015	Error diffusion BTC, " color histogram feature" (CHF), and "bit pattern histogram feature" [11]	Modified Canberra	Corel-1000	Precision: 79.7
			Corel-10000	Precision: 79.8
2018	BTC based on binary ant colony optimization [14]	Modified Canberra	Corel-1000	Precision: 80.565
			Corel-10000	Precision: 65
2008	Quantized HSV color space histogram and dominant color descriptor [5]	L1 distance	Three categories from Corel-1000	Precision: 89.64 Recall: 76.47

2^n, where n is the number of neighborhood pixels considered to generate the codeword. Thus, the local binary pattern descriptor for the pixel (x_c, y_c) can be defined as in the following equations.

$$\mathrm{LBP}(x_c\, y_c) = \sum_{n=1}^{N} f(i_c - i_n)2^n, \tag{4}$$

$$f(x) = \begin{cases} 1, & i_n \geq i_c, \\ 0, & i_n < i_c, \end{cases} \tag{5}$$

where i_c is the value of the center pixel, and i_n is the neighborhood pixel value. $f = 0$ if $i_c \leq i_n$; else, $f = 1$.

LTP [21] is the extension of LBP and resistive to monotonic gray level transformations, in which the threshold k is used to generate the code for the center pixel. If the value of the neighborhood pixel is equal to or larger than the sum of the threshold and center pixel value, the code is +1. If the neighborhood pixel intensity is equal to or if less than the sum of the threshold and center pixel intensity, the code is −1. Else, the code is 0. Thus, the local ternary pattern code for the pixel (x_c and y_c) can be defined as in the following equation:

$$C(p_n, p_c, k) = \begin{cases} 1, & i_n \geq i_c + k, \\ 0, & (i_c - k) < i_n < (i_c + k), \\ -1, & i_n \leq i_c - k, \end{cases} \tag{6}$$

where p_n is the neighborhood pixel; p_c is the center pixel; i_c and i_n are the intensities of the center pixel and neighborhood pixel, respectively. The ternary codeword is then converted into two LBP for designing uniform descriptors by concatenating the histogram of these LBPs.

There are many extensions of LBP proposed in literature, which use local information in various directions such as "Local Tetra Pattern (LTrP)" [22], "Local Binary Extrema Pattern (LBEP)" [28], "Local Derivative Pattern" [29], and "Utilizing multiscale LBP" [30].

Tamura et al. [26] have defined six texture pattern features based on human perception, which can be used to define the image. These features are coarseness, contrast, directionality, line-likeness, regularity, and roughness. The gray level variations and biasness in the distribution of gray levels are measured using the contrast features. The contrast, roughness, and regularity are defined as in equations (7)–(10):

$$\mathrm{Contrast} = \frac{\sigma}{\alpha_4^{1/4}}, \tag{7}$$

$$\alpha_4 = \frac{\mu_4}{\sigma^4}, \tag{8}$$

where μ_4 is the kurtosis, i.e., the fourth moment about mean, and σ^2 is the variance.

In an image, coarseness is the measure of granularity; directionality gives the direction and quality of the edges. It is calculated by convolving the image with Prewitt's horizontal and vertical edge detectors. Line-likeness defines the

average coincidence of direction of edges separated by a pixel distance d. It is constructed by forming the edge direction co-occurrence matrix. Roughness is defined in terms of coarseness and contrast.

$$\text{Roughness} = \text{contrast} + \text{coarseness}. \tag{9}$$

Regularity represents the repetitiveness of patterns. It is defined as

$$\text{Regularity} = 1 - r\left(\sigma_{\text{coarseness}} + \sigma_{\text{contrast}} + \sigma_{\text{directionality}} + \sigma_{\text{linelikeness}}\right), \tag{10}$$

where r is the normalizing factor, and σ^2 is the variance of the respective feature.

In [31], discrete cosine transform (DCT) coefficients are used to represent image texture features as it has an excellent compression capability of energy compaction. The image is divided into subblocks for space localization. DCT is applied to each subblock. The feature vector is calculated using the DC coefficients and some AC coefficients containing the direction-related information. Nine features are extracted from each subblock of size 64. In [32], the Gaussian pyramid is applied to extract the multiresolution images of the R, G, and B planes. DCT is applied on all multiresolution images. The feature vector is generated by concatenating all the DC coefficients and statistical parameters of significant AC coefficients selected through all multiresolution planes. In [33], the color image is converted into a grayscale image. The image is divided into nonoverlapping blocks, and DCT is applied on each block. The histogram is formed using DC coefficients and selected the first three AC coefficients. Six statistical features are calculated using quantization bins for all the blocks.

In [24], the Haar wavelet is used to extract a fixed number of salient points from the image. Gabor texture features and color moments are extracted using the neighborhood pixels of salient points. As the features are extracted for the fixed number of salient points, the computational complexity is better than considering the whole image.

In [34], the discrete wavelet transform is applied on the image up to three scales, and LTrP is used to describe the features of each subband. The artificial neural network is used for image matching and retrieval.

In [25], curvelet transforms are used to find the low-order statistical features of an image. The image and curvelet are transformed to the Fourier domain. The image is then convolved with the curvelet. The curvelet coefficients are calculated by applying the inverse Fourier transform. Standard deviation and the mean of curvelet coefficients represent the image features. Thus, the image is represented by a $2n$ size feature vector, where n indicates the number of curvelet used.

In [35], the ranklet transform is used to generate three images of different orientations, vertical, horizontal, and diagonal, as a preprocessing step for each plane. Ranklet transform is applied on each R, G, and B plane, resulting in the generation of nine images. Each image's standard deviation, mean, and histogram color moments are determined. Thus, a feature vector of size 27 is generated by concatenating all the moments of an image. K-means clustering algorithm is used to cluster images into categories, and the centroid of each category is computed. The query image feature vector is compared with the centroid of each category to find the smallest distance category. All the images that belong to the smallest distance category are compared with the query image for image retrieval.

Table 2 shows a summary of image retrieval techniques based on texture features. Structure-based texture techniques are not suitable for generic image retrieval as the images do not have regular patterns or structures. Statistical texture-based techniques are widely used in generic image retrieval as these techniques are illumination invariant, but the feature vector size is more than other techniques.

4. Shape Features Used in Image Retrieval

Besides texture features and color features, shape features are also used for searching the analogous images as humans observe the objects based on their shape [36–38]. The detailed review of the various shape-based feature extraction and description techniques are presented in [39–41]. Figure 2 shows the various shape-based feature extraction and description techniques. The shape-based features extraction and description techniques can be broadly classified into region-based and contour-based techniques. Contour- or boundary-based techniques basically describe the boundary of the objects, whereas the region-based technique uses all the pixel values of the object. Contour-based methods are categorized as complete object shape-based, if the boundary is represented as a whole shape or primitive/structure-based or if the boundary is segmented into parts and described. Region-based methods are classified as spatial domain-based and transform-domain based. Spatial domain-based techniques are again divided into two types, complete object-based and primitive-based, depending upon the part of the object described.

Shape-based features are not used widely for image retrieval as it requires segmented objects in an image that is challenging to find in heterogeneous dataset images. Generally, the shape-based features are combined with other low-level image features and local features to represent the image for generic applications of image retrieval. Shape-based features are generally used for object retrieval [42, 43].

5. Local Feature Extraction Techniques

The image retrieval techniques can be categorized into local and global techniques. Global image retrieval techniques consider the whole image for extracting and describing. Global feature extraction techniques are suitable to retrieve the duplicate image and can be used for detecting natural scenes. The local feature extraction techniques are useful for detecting human-made objects. In an image, local techniques identify salient regions called interest points or keypoints and express the neighborhood patch of these key points for describing the image. The key points that are

Table 2: Summary of texture feature-based image retrieval techniques.

Year	Method	Similarity measure	Dataset	Performance measure (%)
1997	Subblock DCT DC and AC coefficients [31]	Modified Euclidean distance	200 images of woods, flowers, and sky with mountains	Average retrieval rate (ARR): 82
2017	Multiresolution RGB images, feature vector generated with DCT DC coefficients, and statistical features from the group of AC coefficients [32]	Euclidean distance	Corel-1K	Precision: 87.50 Recall: 17.50 F score: 29.16
			GHIM-10K	Precision: 82.50 Recall: 3.30 F score: 6.35
2013	Quantized histogram statistical texture features generation using DCT with DC and first 3 AC components [33]	Euclidean distance	Corel-1K	Precision: 80 Recall: 81 F score: 80
2000	Wavelet-based salient points, color moments, and Gabor moments of salient points [24]	Mean square error	COREL	Retrieval accuracy: 83.2
2017	LTrP and DWT with the artificial neural network [34]	Artificial neural network	Corel-1K Corel-5k Corel-10K	ARR: 97.9 ARR: 87.42 ARR: 74.13
2008	Curvelet transform with low-order statistics [25]	$L2$ distance	Brodatz texture database	(ARR: 79.54
2012	Ranklet transform and color moments with K-means clustering [35]	Euclidean distance	Wang dataset	Precision: 78.86
2012	Second-order local tetra patterns using vertical and horizontal derivatives of pixels direction [22]	Modified Canberra	Corel-1000	Precision: 75.9 Recall: 48.7
			Brodatz texture database	Recall: 85.3
			MIT VisTex	Recall: 90.02
2012	Edge directional information feature based on local extrema [28]	Modified Canberra	Corel-5000	Precision: 48.8 Recall: 21.1
			Corel-10K	Precision: 40.0 Recall: 15.7
			Corel-1000	Precision:74.8 Recall: 49.16
			Brodatz texture	Recall: 82.68

detected must be highly repeatable so that they can be detected with various transformations such as rotation and illumination. The local features should have the properties such as distinctiveness (high variations), locality (should be small enough to avoid occlusion under different viewing angles), quantity (sufficiently large number of features should be detected for matching purpose), accuracy (feature should be accurately identified in various scales), efficiency (fast to detect), invariance to large deformations, and robustness to small deformations [44]. The commonly used methods for local feature detection techniques are the Harris corner detector [45], Harris–Laplace detector [46], Hessian–Affine detector [47], SURF [48], Shi and Tomasi corner detector [49], difference of Gaussian [50], FAST [51], SUSAN [52], and MSER [53].

The Harris detector uses the edge and corner detector based on autocorrelation function, i.e., a second-moment matrix for local texture description [45]. It finds the intensity differences in all directions for the shift of (u, v) using the following equation.

$$E(p,q) = \sum_{u,v} w(u,v)\left[I(p+u, y+q) - I(u,v)\right]^2. \quad (11)$$

The window function is given as

$$w(u,v) = e^{-\left(u^2+v^2\right)/2\sigma^2}. \quad (12)$$

$E(p, q)$ for small changes shift (p, q) can be expressed as

$$E(p,q) = (p,q)M(p,q)^T, \quad (13)$$

and M is given as

$$M = \sum_{p,q} w(p,q)\begin{bmatrix} IpIp & IpIq \\ IqIp & IqIq \end{bmatrix}, \quad (14)$$

where Ip and Iq are the gradients in p and q direction, respectively; α and β are Eigenvalues of matrix M; then, trace of M is $\alpha + \beta$, and the determinant of M is $\alpha\beta$.

The region is a corner region if CR has a positive value.

$$\mathrm{CR} = (\alpha.\beta) - k.(\alpha + \beta)^2. \quad (15)$$

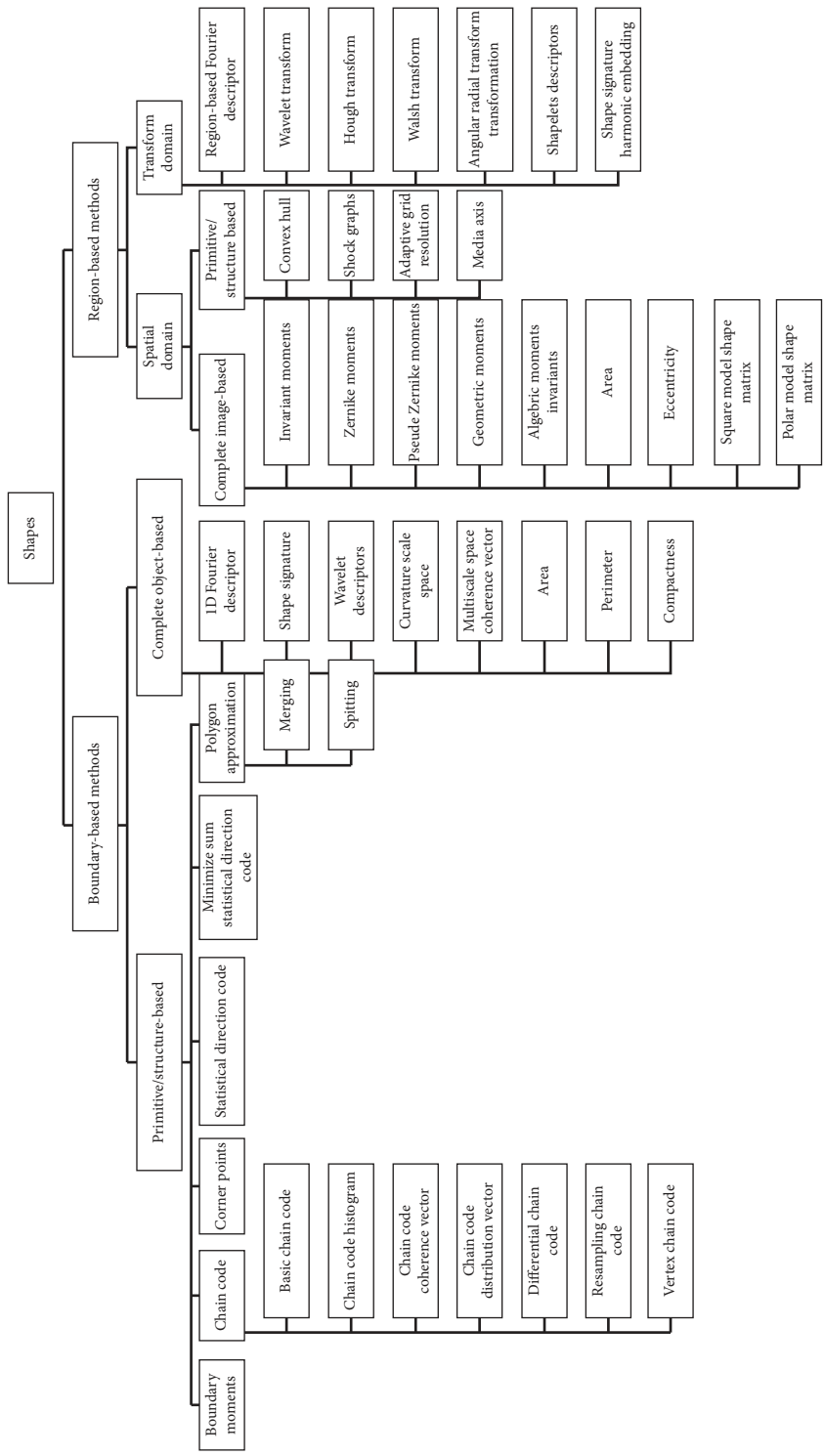

Figure 2: Classification of shape feature extraction and description techniques [39–41].

The Shi and Tomasi corner detector works faster than the Harris corner detector. It works in a similar manner as the Harris corner detector with a slight change in conditions to detect a region as a corner. If the CR value is greater than the threshold, then the region is detected as a corner [49].

$$\mathrm{CR} = \min(\alpha, \beta). \tag{16}$$

The Harris and Shi-Tomasi corner detectors are not so useful for searching similar images of various sizes and scales as they are very unstable to scale change. In [52], a low-level feature detector SUSAN based on a circular mask is presented. SUSAN is fast and accurate to detect lines, edges, and corners with noise reduction. In [54], the Laplacian of Gaussian is approximated with the difference of Gaussian.

The image is made smooth by convoluting it with a Gaussian filter having some width $\sigma 1$. The original image is made smooth by convoluting it with the Gaussian filter having some width $\sigma 2$. The difference between these two Gaussian filter images is taken to find the local features of the image. In [48], Fast Hessian is presented. It uses the Hessian matrix and approximates LoG with a box filter using integral images. It can be applied on multiple scales simultaneously. In [46], Mikolajczyk and Schmid have presented the Harris–Laplace or Hessian–Laplace method of detecting local interest point features at various scales using Laplacian of Gaussian with the Harris corner detector. The points having maximal Laplacian over scales are selected. The selected interest points are not variant to the rotation, scale, and translation.

In [51], FAST, a high-speed machine learning-based corner detection technique, is presented. It uses the segment test criterion, which considers sixteen neighborhood pixels in a circle around a corner candidate keypoint p. It classifies p as a corner if n neighborhood pixels are brighter than $Ip + t$ or darker than the Ip-t. Where Ip is the intensity of candidate keypoint p, and t is the threshold. This technique is very fast as compared to other keypoint detection techniques, but it is not robust to noise and orientation and is dependent upon the threshold value.

The local feature (interest point) can be described with floating-point or binary point descriptors. The detected interest point should be described in a highly differentiable way so that it can be identified and correlated if it exists in some other image. The local feature detection and description techniques need to be elevated for faster retrieval.

One of the most popular local features, the floating-point descriptor, is the scale invariant feature transform (SIFT) that shows the excellent results [50]. SIFT algorithm can be divided in 4 major phases, scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor. SIFT identifies the repeatable features in an image that can be identified in various scales and views by the scale-space function using the difference of Gaussian function between two nearby scale separated images. Local minima and maxima are detected for finding candidate keypoints. Hessian and a derivative of candidate keypoints are performed to reject the noise-sensitive keypoints using detailed fit to the nearby data for location, scale, and the ratio of principal curvatures. Hessian matrix is used to reject the keypoints localized along an edge. A 36 bin orientation histogram is created using the Gaussian-weighted gradient orientation of the keypoint neighborhood pixels by considering the scale at which the keypoint is detected. The dominant direction of local gradients forms the highest peak in the orientation histogram, which is used to create the orientation of the keypoint. These three steps make the image invariant to location, scale, and orientation changes. The keypoint descriptor is formed using the gradient orientations and magnitudes of the keypoint neighborhood pixels by considering the scale and location used to detect the keypoint. The gradient orientations are rotated relative to the keypoint orientation to make them rotation invariant. SIFT provides a floating-point, 128 elements keypoint descriptor. Keypoints between two images are matched by identifying their nearest neighbors with minimum Euclidean distance between descriptors. SIFT shows a good performance in the change of the rotation and scale. It has an excellent performance in images that have a simple background. SIFT represents them without noise. SIFT is good; however, it is not fast enough. To overcome this issue, many variants of SIFT have been proposed such as root-SIFT [55], affine-SIFT [56], color-SIFT [57], edge-SIFT [58], CSIFT [59], NSIFT [60], and PCA-SIFT [61].

In [48], the "Speeded Up Robust Features" (SURF) algorithm is introduced. This keypoint detector and descriptor algorithm is the scale and rotation invariant. It is based on the Hessian matrix and approximates LoG with a box filter using integral images and can be applied on multiple scales simultaneously. It uses the Hessian matrix determinant for calculating the keypoint location and scale. It employs Haar wavelet responses for orientation assignment and description. Haar wavelet responses are represented as a vector with a total of 64 dimensions. SURF is not affine invariant.

In [62], BRIEF, a binary descriptor, is presented. The image block descriptor is calculated by taking a simple intensity difference between pixels of an image block. The BRIEF keypoint feature descriptor is suitable for real-time applications due to its speed. However, it has a low tolerance for transformations such as rotation, scale, and image distortions. In [63], a binary keypoint detection, description, and matching technique, BRISK is presented. In this, keypoints are detected with a scale-space pyramid consisting of octaves and intraoctaves. It uses a 9–16 mask of the FAST feature detection technique, which requires at least 9 pixels to be lighter or darker than the center pixel and FAST 5–8 mask on octave c0 to obtain the FAST scores. The keypoint detected is described with a bit string of length 512 in a binary format by considering the results of the brightness comparison test with the direction of keypoint to make it rotation invariant. The BRISK keypoint descriptors are fast and easy to match, since the simple Hamming distance is calculated between them. Hamming distance gives the dissimilarity between matched keypoints. In [64], the "Oriented FAST and Rotated BRIEF-(ORB-)" based binary feature descriptor is presented. It is quicker to compute as compared to SURF and SIFT but has limitations in the descriptive power and scale invariance in some situations. In [65], a binary feature descriptor inspired by the human eye behavior "Fast Retina Keypoint (FREAK)" is presented. The human eye uses the difference of Gaussian to extract the features from an image at various sizes and encodes them. It uses the retina sampling grid. It is circular in nature with high-density points near the center. The sample points are made smooth to remove the noise with different kernel sizes. The receptive fields are overlapped to capture more information and improve the discriminative power and performance. The descriptor is created by the one-bit encoding of the difference between the receptive fields and the Gaussian kernel. The receptive fields which are selected should be uncorrelated or low correlated and highly discriminate. Thus, the difference of Gaussian from coarse to fine ordering is selected. Initially, the FREAK descriptor's first 16 bytes are compared for matching the keypoints. If the distance

between the first 16 bytes is less than the threshold, the next bytes are compared. Thus, searching is also performed from the coarse to a fine level. Matching the first 16 bytes increases the speed of matching. The rotation of keypoint is calculated using the sum of local gradients.

The local feature detectors can be compared on the basis of image contents and structures such as corners, blobs, or regions and discriminative powers with respect to various invariances [66]. Table 3 shows a comparison of various local feature detection techniques. The selection of local feature detection is completely based on the type of images in the dataset. The local feature detection techniques are still not highly robust to the scale and affine transformations and have limited repeatability and robustness properties.

Table 4 shows the average retrieval accuracy for the augmented Wang dataset using various feature extraction and description techniques. The augmented Wang dataset contains 1100 images of 11 different categories. The existing techniques are reimplemented and tested on the datasets to normalize the test environment. Mean square error is used as the distance measure. BRISK, ORB, FAST, MSER, and SURF are used for interest point detection. Feature descriptors such as BRISK, FREAK, SURF, and ORB are used. The average retrieval accuracy value is 1.11% when BRISK is used for feature extraction and description. When features are extracted with BRISK and described using FREAK, the retrieval accuracy is 6.08%. The average retrieval accuracy is 3.62% when ORB is used. With MSER as a feature extractor and SURF as a feature descriptor, the retrieval accuracy is 15.65%. When FAST is used for feature detection and FREAK is used for description, average retrieval accuracy is 12.61%. When SURF is used as a feature detector and FREAK is used as a feature descriptor, the average retrieval accuracy is 12.16%. The highest average retrieval accuracy is 22.10% when SURF is used as a feature detector and descriptor.

The floating-point descriptors such as SURF have high retrieval accuracy but have high memory requirements and not suitable for real-time applications. The binary descriptors are good for fast matching, computation, and low memory requirements but face the issues of low descriptive power, robustness, and generality.

6. Feature Fusion-Based Techniques Used in Image Retrieval

The image datasets contain images that are highly diverse and nonhomogeneous in nature. It is very difficult to retrieve the images by using simple and individual low-level image features. Therefore, in literature, the performance of image retrieval systems has been enhanced by combining the low-level features (color, shape, and texture), global features, and local features for representing the feature vectors.

In [67], the image shape and color features are merged to generate hybrid image features. Color moments mean, standard deviation, and skewness are extracted as color features, and seven invariant moments are extracted as shape features from the second and third moments to represent an image. These features of the query image are compared with dataset images using $L2$ similarity measure. For performance evaluation, precision and recall parameters are used.

In [68], image features are generated using texture and color features. HSV color moments, i.e., mean, skewness, and standard deviation, are calculated. Texture features are generated using a 2D Gabor filter by varying the scale and rotation. Euclidean distance is used as a similarity measure. Precision is used as a performance metric.

In [69], scale and illumination robust feature vectors are generated by the fusion of texture and color features. Here, multilevel Haar wavelet features are combined with a color histogram to increase retrieval accuracy.

In [70], color and edge features are combined to generate a robust color volume histogram-based feature vector. The HSV color space image is generated from an RGB color image. The H, S, and V components are uniformly quantized into 72 bins. Sobel edge detection operator is applied to the V component to generate a quantized edge map of the image containing 32 bins. $L1$ distance is used as similarity measurement criteria.

In [71], the local feature descriptor technique is combined with a bag of words. SURF- and SIFT-based local features are generated. K-means clustering is used to generate visual words from these extracted features. Images that match the query image are retrieved from the dataset using the SVM classifier. For performance evaluation, precision and recall parameters are used.

In [72], the color contents, shape, and color texture are used for generating the features of the image. The color contents are extracted by calculating the summation of median and variance from the histogram of each plane R, G, and B. Features are made independent of rotation and illumination by extracting shape features. The RGB image is first converted to a grayscale image. Salt and pepper noise is removed by applying the median filter. An image feature vector based on the shape is generated by applying the neutrosophic clustering algorithm and the Canny edge detection algorithm. The texture and color feature's standard deviation, mean, contrast, energy, and homogeneity are calculated based on horizontal, vertical, and diagonal directions, using GLCM by applying the Gaussian filter and dividing the image into 4×4 blocks. All these features are stored in the database as feature vectors. For similarity measurement and retrieval of matching images, memetic algorithm based on genetic and great deluge algorithm are used.

In [73], chromaticity moments, co-occurrence, and color moments features are fused to generate a feature vector. Shape and distribution chromaticity moments are calculated using CIE xyY color space. For each color plane, standard deviation, mean, and skewness color moments are determined using RGB color space. Contrast, energy, homogeneity, correlation, and entropy color co-occurrence statistical features are calculated using RGB color space. Inverse variance weighted Euclidean distance is used as a similarity measure to improve accuracy.

In [74], local feature extraction techniques are combined with a bag of visual words (BoVW). For each image in the training dataset, the SURF and FREAK features are calculated. K-means++ clustering algorithm is used to reduce

Table 3: Comparison of various local feature detection techniques [66].

Local feature Detector	Image structure detected	Invariance					Property			
		Translation	Illumination	Rotation	Scale	Affine	Repeatability	Localization	Robustness	Efficiency
Harris and Stephens [45]	Corner	Yes	Yes	Yes	No	No	High	High	High	Average
Hessian	Blob	Yes	Yes	Yes	No	No	Average	Average	Average	Low
SUSAN [52]	Corner	Yes	Yes	Yes	No	No	Average	Average	Average	High
Harris-Laplace [46]	Corner and blob	Yes	Yes	Yes	Yes	No	High	High	Average	Low
Difference of Gaussian [50]	Corner and blob	Yes	Yes	Yes	Yes	No	Average	Average	Average	Average
SURF [48]	Corner and blob	Yes	Yes	Yes	Yes	No	Average	High	Average	High
SIFT [50]	Corner and blob	Yes	Yes	Yes	Yes	No	Average	High	High	Average
MSER [53]	Region	Yes	Yes	Yes	Yes	Yes	High	High	Average	High

Table 4: Average image retrieval accuracy using various feature extraction and description techniques for the augmented Wang dataset.

Category	Feature extractor-feature descriptor						
	BRISK-BRISK [63]	BRISK-FREAK [63, 65]	ORB-ORB [64]	MSER-SURF [48, 53]	FAST-FREAK [51, 65]	SURF-FREAK [48, 65]	SURF-SURF [48]
Tribe	1.09	7.32	2.41	14.39	15.84	24.55	22.18
Beach	1	3.35	1.27	14.81	9.18	18.37	16.33
Monuments	1.12	6.3	1.49	15.37	11.26	12.36	13.29
Bus	1	7.74	1.6	21.2	21.3	10.55	37.77
Dinosaurs	1.17	14.6	22.95	28.8	27.48	26.58	57.25
Elephant	1.25	5.35	2.25	13.63	11.88	7.79	17.14
Roses	1	3.66	1.72	15.82	7.4	10.92	21.5
Horses	1.32	3.24	1.92	15.77	8.43	5.85	13.51
Mountains	1.01	5.25	1.22	9.9	6.13	5.19	11.2
Food items	1.02	5.74	1.31	8.2	6.03	5.3	11.37
Aeroplane	1.24	4.38	1.71	14.29	13.74	6.3	21.61
Average retrieval accuracy (%)	**1.11**	**6.08**	**3.62**	**15.65**	**12.61**	**12.16**	**22.10**

feature vector space, and clusters are generated. Each visual word represents the center of the cluster, and it is used to generate the codebook or vocabulary. The visual words of FREAK and SURF are fused together by concatenation. A histogram is constructed for each image of the dataset and given to the support vector machine (SVM) as the input. The Euclidean similarity measure method is used to calculate the similarity score of the query image and dataset image collection.

In [75], local and global features are fused together by considering the SURF and histogram of gradient (HoG) features of the image. The SURF and HoG features are obtained from the image. Visual words vocabulary using the K-means algorithm is generated from training image features. The histogram of visual words is input to train the SVM Hellinger kernel function. Euclidean distance is utilized to retrieve the images from the image dataset collection.

In [76], "Weighted Average of Triangular Histograms (WATH)" of visual words are considered to add spatial content information of the image. This helps in reducing two problems: first, interpretation gap issues due to low-level image features and high-level image semantic and second, overfitting problem due to the large visual dictionary.

In [77], image retrieval based on the multiregion has been presented using the curvelet transform and color features of significant regions. There are three major steps involved: important regions identification from RGB images, representation of regions using several features, and retrieval of relevant images using regions from query and target images from the dataset. The regions which engage users' attention are called important regions. An image can have multiple important regions. Important regions are extracted using a saliency map, location, size, and region homogeneity. The hue component is used to find the significant regions of the image. Significant region is represented using histogram-based color descriptors. The RGB image is converted into HSV color space. The hue component is divided into 16 bins. S and V components are divided into four bins each. Twenty-four features are extracted from each significant region. The texture feature descriptors of each significant region are computed using the curvelet transform. The histogram intersection technique is applied to measure the color closeness between images. The texture closeness is computed using Euclidean distance. The total distance is the summation of the distance between the color feature and texture feature of the query image and dataset image. The system is evaluated using precision, recall, and Fmeasure.

In [78], authors have presented Sphere/Rectangle Tree indexing and locality sensitive hashing techniques with bag of visual words. SURF is used to describe the image features. Images visual vocabulary is created using bag of visual words (BoVW). Locality sensitive hashing, Sphere/Rectangle Tree, $L1$ norm (Manhattan distance), and $L2$ norm (Euclidean distance) are used to find the nearest visual words of the query image's vector.

In [79], authors have presented the technique to combine texture, edge, and color features of the image. It uses modified color difference histogram features in Lab color space to extract texture and color features. The edge orientation features are calculated in Lab color space using the Sobel operator. Query execution complexity is reduced by stagewise execution. Initially, similar images are selected based on color features. From this selected set of images, texture features are compared and given as an input for edge feature matching, and finally, similar images are retrieved. Precision, recall, and bull's eye performance measures are used to evaluate the method.

In [80], authors have presented a technique to fuse the texture and color features. The color features are generated using a histogram of the quantized HSV color space image. Texture features are generated using GLCM, LBP, and normalized moment of inertia (NMI). NMI is calculated using the particle swarm optimization-based pulse code neural network (PCNN) and 2D Otsu image segmentation method. The technique fuses these features based on the weight assigned to each feature.

In [1], authors have presented a method to fuse the texture features and color features. The RGB color space

intensity values of the pixels are combined with quantized HSV color space values. The V component values are used to find the quantized edge and intensity information. An extended weighted $L1$ distance is used for similarity matching.

In [81], the image retrieval technique based on the fusion of color, texture, and shape features is presented. The color moments average, standard deviation, skewness, and kurtosis are calculated for each plane of the color image. The gray image is used to find the texture feature by dividing the image into 8×8 blocks and applying DCT on it. The feature vector is created using DC components and specific AC components. For generating the shape features, the image is first segmented into salient regions by applying the c-means algorithm using color and texture feature vectors. Then, the principal axis of the region is calculated. The shape feature vector is generated using the endpoints of the principal axis. Matching images are retrieved by using SVM from the dataset.

Table 5 shows a summary of various feature fusion-based techniques reviewed. Significant work is carried out in fusing color features with texture features or local features. More focus is given on color features in RGB or HSV color space combined with SURF-based local features.

Local feature extraction techniques generate discriminative features based on corners, edges, and blobs. These techniques work well for the images containing objects, monuments, and artifacts. Features are detected from occluded object images also. These are invariant to scaling, translation, and illumination conditions. But these techniques do not work well for scenic images such as images of forest, mountains, beaches, and sky, as the regions contain a large number of corners and edges. These methods do not work well for objects that are smooth in texture and does not have many edges or corners. Global features or low-level feature extraction techniques work well for such images but are not suitable for occluded objects and illumination variation conditions. Thus, fusion techniques that combine the local features with low-level features based on color and statistical features give a better performance as compared to other techniques studied.

7. Datasets for Image Retrieval Used in Experimentation

Various datasets are available in the literature. These datasets contain the images of human-made objects, natural objects, buildings, landmarks, animals, natural scenes of beaches, mountains, and water. The images are taken with variations in conditions of illumination, rotation, scaling, and occlusion. The commonly used datasets for image retrieval are Flickr Logos 27, FlickrLogos-32, Flickr1M, Amsterdam Library of Object Images (ALOI), UKBench, INSTRE, ZuBuD, Corel-1000, COIL, Caltech 101, and Caltech-256. Figure 3 shows some of the images of these datasets.

ALOI dataset [83] contains 110250 color images in the PNG format of 1000 small objects with more than 100 images per object. Each object is captured by changing the viewing angle in 72 directions, illumination direction with 24 configurations, and illumination color with 12 configurations. Each image is of size 768×576 pixels.

COIL dataset [84] contains 7200 images of 100 objects with 72 images per class. These images were captured by placing the object on a turntable with a black background. The images are in the PNG format, with a size of 128×128 pixels.

UKBench [85] dataset contains 10200 images of 2550 classes, with four images per class. The images are blurred and rotated. Each image is of size 640×480 pixels. All the images can be used as query images.

Stanford Mobile Visual Search [86] dataset contains images clicked using various camera phones. The images are of size 640×480 pixels with varying distortions and illumination conditions of objects such as text documents, landmarks, CD covers, books, and paintings. The images are categorized into 1200 categories with 3300 query images.

Holiday [87] dataset contains 1491 of 500 classes, with scenic holiday images of nature, water, fire, and human-made objects with changes in rotation, viewpoint, illumination, and blurring. The first image of each class is the query image.

Oxford-5K [88] dataset contains 5062 images of 11 Oxford building landmarks with 55 query images and some distracter images. The images are of size 1024×768 pixels. These images are downloaded from Flickr and manually annotated. Paris dataset [89] contains 6412 images of 12 Paris landmarks such as the Eiffel Tower, Hotel des Invalides, Moulin Rouge, La Defense, Louvre, Notre Dame, Musee d'Orsay, Pantheon, Pompidou, Sacre Coeur, and Arc de Triomphe collected from Flickr with 500 query images. The images are of size 1024×768 pixels. ZuBuD dataset [90] contains 1005 images of 201 Zurich buildings with five viewpoints. Each image is of size 320×240 pixels. There are 115 query images with varying viewpoints and illumination conditions.

Flickr Logos 27 dataset [91] is a labeled dataset created using downloaded logo images of brands such as Adidas, BMW, Coca-Cola, Pepsi, Vodafone, FedEx, DHL Intel, Google, Nike, and Puma from Flickr. There is a total of 27 classes and 30 images per class in the training dataset. There is a total of 270 images in the query dataset with five images per class and 135 images that do not belong to any class. FlickrLogos-32 [92] dataset contains images of 32 logo brands of size 1024×768 pixels. These images were downloaded from Flickr. The images are partitioned into training, validation, and query set. Out of 8240 images present in the dataset, 6000 are distracter images. Flickr1M dataset [93] consists of 1197398 images downloaded from Flickr. The image categories are broadly classified as objects (such as bicycle, birds, chairs, cats, tables, and trees), landmarks (such as Golden Gate Bridge, Tower Bridge, and Colosseum), scenes (such as the beach, city, people, sunset, and desert), and activities (such as baseball, sailing, sailboat, Christmas, and wedding).

INSTRE [94] is an object dataset. The dataset is divided into three datasets: INSTRE-S1 of 100 classes for single object case with 11011 images, INSTR-S2 of 100 classes for single object case with 12059 images, and INSTRE-M for

Table 5: Summary of feature fusion-based image retrieval techniques.

Year	Method	Similarity measure	Dataset	Performance measure (%)
2017	A fusion of color moments and seven invariant moments [67]	Euclidean distance	Wang	Precision: 66.2
2017	A fusion of HSV color moments and the Gabor filter-based texture features [68]	Euclidean distance	Wang	Precision: 65.6
2017	A fusion of color histogram features and multilevel Haar wavelet-based texture features [69]	Euclidean distance	Wang	Objective computations not given
2019	Color volume histograms using quantization of HSV color space and edges [70]	$L1$ distance	Corel-5000	Precision: 60.13 Recall: 7.21
			Corel-10000	Precision: 48.58 Recall:5.83
2015	Fusion of SIFT with BoVW [71]	$L2$ distance	ALOI	Precision: 88 Recall: 29
			Flickr	Precision: 78 Recall: 26
2017	A fusion of color histogram features and shape features using a Canny edge detector [72]	Threshold-based	Corel-1K	Precision: 88.2 Recall:70.02
2018	Chromaticity color moments fused with statistical features of color co-occurrence [73]	Euclidean distance (weighted)	Wang	Precision: 83.83 Recall: 10.1
2018	SURF descriptors fused with HoG feature descriptors [75]	Euclidean distance	Corel-1000	Precision: 80.61
			Corel-1500	Precision: 76.28 Recall: 15.25
			Corel-5000	Precision: 60.60 Recall: 12.12
			Caltech-256	Precision: 46.30 Recall: 09.26
2018	Fusion of the SIFT descriptor with BoVW [76]	Euclidean distance	Corel-1000	Precision: 87.85 Recall: 17.37
			Corel-1500	Precision: 84.38 Recall: 16.88
2014	A fusion of color features and curvelet features [77]	Euclidean distance	Corel-1000	Precision: 81
2018	SURF- and FREAK-fused feature descriptors using BoVW [74]	Euclidean distance	Corel-1000	Precision: 86.00 Recall: 17.19
			Corel-1500	Precision: 83.20 Recall: 16.64
			Caltech-256	Precision: 38.98 Recall: 7.796
2019	Modified color difference histogram [79]	Euclidean distance	Wang subset	Precision:75.33 Recall: 18.61 Bull's eyes Percentage:48.74
2019	A fusion of HSV color space, GLCM, LBP, and normalized moment inertia [80]	Euclidean distance	Corel-1k	Accuracy:69.7 Recall: 69.1 *F*measure: 69.4
			AT&T face dataset	Accuracy: 74.6 Recall: 70.9 *F*measure: 72.7
			FD-XJ face dataset	Accuracy: 62.9 Recall: 61.8 *F*measure: 61.8
2020	Intensity variation descriptor by fusion of HSV and RGB color features, edges, and intensity variations-based texture features [1]	Extended $L1$ distance	Corel-5K	Precision: 66.9 Recall: 8.03 *F*measure: 14.34
			Corel-10K	Precision: 56.88 Recall: 6.83 *F*measure: 12.20

Table 5: Continued.

Year	Method	Similarity measure	Dataset	Performance measure (%)
2017	Fusion low order color moments, DCT-based texture features, and salient region-based shape features with the SVM classifier [81]	Normalized matching ratio	Corel-1000	Precision 78.1 Recall 17.2
			Oxford flowers	Precision 82.3 Recall 18.7
			Caltech-256	Precision 47.0 Recall 20.3
2017	A fusion of color moments, HSV histogram, co-occurrence matrix, wavelet moments, and chain code features [82]	Manhattan distance	Wang	Precision: 82.4

Figure 3: Sample images from standard datasets.

multiple object case with two different objects in each image with total 5473 images. The images are broadly divided into three categories—architectures such as buildings, planar objects such as paintings and designs, and daily stereoscopic objects such as toys and products.

Caltech 101 [95] contains 9144 images of 101 objects categories and one clutter category with a minimum of 31 images per category that are roughly of size 300×200 pixels. Caltech-256 [96] containing 30608 images of 256 object categories and one clutter category with a minimum of 80 images per category. The images can be classified into two broad categories of animated and unanimated images.

Corel-1000 [97, 98] dataset contains 1000 images of 10 classes and 100 images per class. Each image is of size 256×384 pixels and in the JPEG format. The images are from categories such as dinosaurs, elephants, horses, flowers, mountains, beach, food items, and bus.

8. Similarity Measures Used in Image Retrieval

There are many distance metrics or similarity measures defined in the literature to compare the query image with the images in the dataset such as Manhattan distance, Euclidean distance, Chebychev distance, Minkowski distance, cosine distance, square chord distance, fidelity distance, Sorensen distance, Canberra distance, squared chi-squared distance, and Mahalanobis distance [15, 99–102]. The distance metric uses a distance function to compare the images. It is selected depending upon the features that are used for representing the image. The distance metrics discussed here use numerical or continuous values.

Minkowski distance (Lp distance) between two feature vectors q and d is the p^{th} square root of the sum of p^{th} power of the absolute difference between the image feature vector pair as given in the following equation. If $p = 1$, it is called as

city block distance. If $p = 2$, it is called as Euclidean distance. Minkowski distance is a homogeneous and translation invariant metric as it uses normed vector space.

$$\mathrm{MD}(q,d) = \sqrt[p]{\sum_{i=1}^{n} |q_i - d_i|^p}. \tag{17}$$

Manhattan distance ($L1$/city block/taxicab distance) between two feature vectors is given by the sum of the absolute difference between each vector dimension pair as given in the following equation. It is used to calculate the distance in a grid-like path between two feature vectors. This metric is robust to outliers, but it is sensitive to variations in the background such as color, illumination, light direction, and size.

$$\mathrm{MD}(q,d) = \sum_{i=1}^{n} |q_i - d_i|. \tag{18}$$

Extended $L1$ distance [1] is the distance between the query image and dataset image feature given by

$$EL1(q,d) = \sum_{i=1}^{n} \frac{|d_i - q_i|}{|q_i + d_i + w * u_q|}, \tag{19}$$

where

$$u_q = \sum_{i=1}^{n} \frac{d_i}{n}. \tag{20}$$

Euclidean distance ($L2$ distance) is the square root of the summation of the square difference between each vector dimension pair of q and d as given in the following equation. It can be used to calculate the distance between two data points in a plane. This distance metric is most commonly used for similarity measurement in image retrieval because of its efficiency and effectiveness.

$$ED(q,d) = \sqrt{\sum_{i=1}^{n} (q_i - d_i)^2}. \tag{21}$$

Mean square error is the mean of the sum of the square difference between each vector dimension pair as given in the following equation. The computational complexity of mean square error is more than the sum of absolute difference as the square of differences is calculated. Distance is always a large positive number. It can be used in both spatial and transform domain images.

$$MSE(q,d) = \frac{1}{n} \sum_{i=1}^{n} (q_i - d_i)^2. \tag{22}$$

Chebychev distance, also called L∞ distance, is given by the following equation. The distance between image feature vectors is calculated as the maximum absolute difference between the pair of features of the image.

$$\mathrm{ChD}(q,d) = \max_i |q_i - d_i|. \tag{23}$$

Square chord distance is defined as the sum of the squared difference between the square roots of the image feature vector dimension pair as given in the following equation.

$$SCD(q,d) = \sum_{i=1}^{n} \left(\sqrt{q_i} - \sqrt{d_i} \right)^2. \tag{24}$$

Fidelity distance is the summation of the square root of the product between Q and D feature vector dimension pair as given in the following equation.

$$FD(q,d) = \sum_{i=1}^{n} \left(\sqrt{q_i d_i} \right). \tag{25}$$

Sorensen distance is the summation of the absolute difference divided by summation of absolute addition between the feature vector dimension pair as given in the following equation.

$$SD(q,d) = \frac{\sum_{i=1}^{n} |q_i - d_i|}{\sum_{i=1}^{n} |q_i + d_i|}. \tag{26}$$

Canberra distance is the summation of the absolute difference between feature vector dimensions pair divided by the addition of the absolute value of the feature vector dimension pair as given in the following equation. This method is useful for the data spread about the origin. This method is similar to the city block distance metric. City block distance gives larger values between dissimilar images. Canberra distance normalizes this by dividing it with the sum of the feature pairs.

$$CD(q,d) = \sum_{i=1}^{n} \frac{|q_i - d_i|}{|q_i| + |d_i|}. \tag{27}$$

Modified Canberra is the summation of the absolute value of the difference between feature vector dimensions pair divided by the addition of feature vector dimension pair as given in the following equation.

$$MCD(q,d) = \sum_{i=1}^{n} \left| \frac{q_i - d_i}{q_i + d_i + \varepsilon} \right|. \tag{28}$$

Squared chi-squared distance is the summation of the squared difference between feature vectors divided by the absolute addition of the feature vectors given in the following equation.

$$SCSD(q,d) = \sum_{i=1}^{n} \frac{(q_i - d_i)^2}{|q_i + d_i|}. \tag{29}$$

Mahalanobis distance is a quadratic form distance metric where Σ^{-1} is a covariance matrix of feature vector q and d as given in the following equation (30). It measures the similarity between two feature vectors by taking covariance into consideration. It is used for calculating distance in multivariate space.

$$MD(q,d) = \left[(q-d)^T \sum^{-1} (q-d) \right]^{1/2}. \tag{30}$$

Cosine distance provides the angular difference between the feature vectors as given in the following equation. This metric is generally used when the orientation between the feature vectors is important and the magnitude does not matter.

$$CD(q,d) = 1 - \frac{|qd'|}{\sqrt{(qq')(dd')}}. \quad (31)$$

9. Performance Evaluation of Image Retrieval

The quality of the image retrieval methods can be evaluated by the accuracy of the method based on the rank assigned to the images retrieved. The retrieved images can be classified as relevant and irrelevant images. In literature, recall and precision have been widely used to evaluate the quality of the image retrieval methods. The precision for a query image is defined as in equation (32). Recall for a query image is defined as in equation (33).

$$\text{Pr} = \frac{\text{relevant images retrieved count}}{\text{total images retrieved count}}, \quad (32)$$

$$\text{Rc} = \frac{\text{relevant images retrieved count}}{\text{total relevent images in dataset count}}. \quad (33)$$

Precision and recall are combined to find the *F*score/*F*measure to compute the performance of the retrieval method. *F*score is defined as in equation (34).

$$F\ \text{score} = \frac{w * \text{Pr} * \text{Rc}}{\left(w^2 * \text{Pr}\right) + \text{Rc}}, \quad (34)$$

where w is the parameter used to give weightage to precision over recall.

10. Conclusion and Possible Future Research Directions in Image Retrieval

In this article, a detailed review of CBIR-based techniques proposed in the last 10–15 years, using various feature extraction and description techniques, has been presented. Low-level features are used to represent images with texture features, shape features, and color features. The standard dataset images are diverse and complex in nature due to rotation, translation, scale, and affine variances. Therefore, one type of low-level feature cannot represent the image with high discriminative power. The fusion of multiple low-level feature representations can enhance the performance of the retrieval system. The global feature extraction techniques work well with nature's scenic images, but give less performance in the case of images containing human-made structures and objects. In the case of local feature extraction techniques, features are extracted from the image regions located near the interest point instead of the complete image, and thus work well for partially visible objects. These techniques are not suitable for nature's scenic images, as many local features are extracted. The fusion of low-level image features with local features can improve the performance of the system. Blob-based SURF variant and region-based MSER variant techniques can be fused with texture features and color features for improving the accuracy of the system. Local feature extraction techniques generate large feature descriptors of varying sizes. The size of the feature descriptor needs to be converted into optimal length so that the speed of query execution can be improved. The fusion of machine learning algorithms with local image features, low-level image features, and statistical features might improve the performance of the system. Image retrieval using deep neural network-based algorithms gives better results as compared to the traditional local and global feature description techniques but requires high computing power and fine-tuning of the network. The low-level features and local features require less computing power. The fusion of these two methods is a possible research area. The performance of the image retrieval techniques can be improved by combining the other clues such as image annotations, web search history, text in the web pages, and speech present in the videos. The standard datasets that are used currently for image retrieval techniques have been designed majorly for image classification. There is a need for datasets specifically developed for image retrieval with a large number and categories of images.

Data Availability

The data used to support this study are available in the form of earlier published literature.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Z. Wei and G.-H. Liu, "Image retrieval using the intensity variation descriptor," *Mathematical Problems in Engineering*, vol. 2020, Article ID 6283987, 12 pages, 2020.

[2] S. Singh and S. Batra, "An efficient bi-layer content based image retrieval system," *Multimedia Tools and Applications*, vol. 79, no. 25-26, p. 17731, 2020.

[3] W. Zhou, H. Li, and Q. Tian, *Recent Advance in Content-Based Image Retrieval: A Literature Survey*, Cornell University, Ithaca, NY, USA, 2017.

[4] J. Han and K. K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 11, no. 8, pp. 944–952, 2002.

[5] H. Shao, Y. Wu, W. Cui, and J. Zhang, "Image retrieval based on MPEG-7 dominant color descriptor," in *Proceedings of the 2008 the 9th International Conference For Young Computer Scientists*, pp. 753–757, Hunan, China, November 2008.

[6] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the Fourth ACM International Conference on Multimedia*, pp. 65–73, Boston, MA, USA, November 1996.

[7] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proceedings*

Of IEEE Computer Society Conference On Computer Vision And Pattern Recognition, pp. 762–768, San Juan, PR, USA, June 1997.

[8] E. Delp and O. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, 1979.

[9] H. B. Kekre, S. D. Thepade, and A. T. Lohar, "Image retrieval using block truncation coding extended to color clumps," in *Proceedings of the 2013 International Conference on Advances in Technology and Engineering (ICATE)*, pp. 1–6, Mumbai, India, February 2013.

[10] J.-M. Guo, H. Prasetyo, and N.-J. Wang, "Effective image retrieval system using dot-diffused block truncation coding features," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1576–1590, 2015.

[11] J.-M. Guo, H. Prasetyo, and J.-Ho Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 466–481, 2015.

[12] J.-M. Guo and Y.-F. Liu, "Improved block truncation coding using optimized Dot diffusion," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1269–1275, 2014.

[13] J.-M. Guo and H. Prasetyo, "Content-based image retrieval using features extracted from halftoning-based block truncation coding," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1010–1024, 2015.

[14] Y.-H. Chen, C.-C. Chang, C.-C. Lin, and C.-Y. Hsu, "Content-based color image retrieval using block truncation coding based on binary ant colony optimization," *Symmetry*, vol. 11, no. 1, p. 21, 2018.

[15] S. R. Badre and S. D. Thepade, "Novel video content summarization using thepade's sorted n-ary block truncation coding," *Procedia Computer Science*, vol. 79, pp. 474–482, 2016.

[16] S. D. Thepade, R. K. K. Das, and S. Ghosh, *Image Classification Using Advanced Block Truncation Coding with Ternary Image Maps*, pp. 500–509, Springer, Berlin, Germany, 2013.

[17] N. V. Soniminde and S. D. Thepade, "Global windowing based thepade's sorted N-ary block truncation coding (TSnBTC) for content based video retrieval with various similarity measures," in *Proceedings of the 2018 Fourth International Conference On Computing Communication Control And Automation (ICCUBEA)*, pp. 1–6, Pune, India, August 2018.

[18] R. Das, S. Thepade, S. Bhattacharya, and S. Ghosh, "Retrieval architecture with classified query for content based image recognition," *Applied Computational Intelligence and Soft Computing*, vol. 2016, Article ID 1861247, 9 pages, 2016.

[19] J. Wang, L. Wang, X. Liu, Y. Ren, and Y. Yuan, "Color-based image retrieval using proximity space theory," *Algorithms*, vol. 11, no. 8, p. 115, 2018.

[20] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[21] X. Tan and B. Triggs, "enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[22] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local Tetra patterns: a new feature descriptor for content-based image retrieval," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2874–2886, 2012.

[23] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[24] E. Loupias, N. Sebe, S. Bres, and J.-M. Jolion, "Wavelet-based salient points for image retrieval," in *Proceedings 2000 International Conference On Image Processing (Cat. No. 00CH37101)*, pp. 518–521, Vancouver, Canada, September 2000.

[25] I. J. Sumana, M. M. Islam, D. Zhang, and G. Lu, "Content based image retrieval using curvelet transform," in *Proceedings of the 2008 IEEE 10th Workshop on Multimedia Signal Processing*, pp. 11–16, Cairns, Australia, October 2008.

[26] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[27] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[28] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Directional local extrema patterns: a new descriptor for content based image retrieval," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 191–203, 2012.

[29] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 533–544, 2010.

[30] P. Srivastava and A. Khare, "Utilizing multiscale local binary pattern for content-based image retrieval," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12377–12403, 2018.

[31] H.-J. Bae and S.-H. Jung, "Image retrieval using texture based on DCT," in *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat. No.97TH8237)*, pp. 1065–1068, Singapore, September 1997.

[32] N. Varish, S. Kumar, and A. K. Pal, "A novel similarity measure for content based image retrieval in discrete cosine transform domain," *Fundamenta Informaticae*, vol. 156, no. 2, pp. 209–235, 2017.

[33] F. Malik and B. Baharudin, "Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain," *Journal of King Saud University-Computer and Information Sciences*, vol. 25, no. 2, pp. 207–218, Jul. 2013.

[34] M. Sadafale and S. V. Bonde, "Spatio-frequency local descriptor for content based image retrieval," in *Proceedings of the 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, pp. 1–5, Kollam, India, August 2017.

[35] A. J. Afifi and W. M. Ashour, "Image retrieval based on content using color feature," *ISRN Computer Graphics*, vol. 2012, Article ID 248285, 11 pages, 2012.

[36] X. Wang and K. Xie, "A novel direction chain code-based image retrieval," in *Proceedings of the Fourth International Conference On Computer And Information Technology*, pp. 190–193, Los Alamitos, CA, USA, September 2004.

[37] F. Baji and M. Mocanu, "Chain code approach for shape based image retrieval," *Indian Journal of Science and Technology*, vol. 11, no. 3, pp. 1–17, 2018.

[38] J. Sun and X. Wu, "Chain code distribution-based image retrieval," in *Proceedings of the 2006 International Conference On Intelligent Information Hiding And Multimedia*, pp. 139–142, Pasadena, CA, USA, December 2006.

[39] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[40] A. Amanatiadis, V. G. Kaburlasos, A. Gasteratos, and S. E. Papadakis, "Evaluation of shape descriptors for shape-based image retrieval," *IET Image Processing*, vol. 5, no. 5, p. 493, 2011.

[41] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee, "Shape measures for content based image retrieval: a comparison," *Information Processing & Management*, vol. 33, no. 3, pp. 319–337, 1997.

[42] H. Zhang, Z. Dong, and H. Shu, "Object recognition by a complete set of pseudo-Zernike moment invariants," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 930–933, Dallas, TX, USA, March 2010.

[43] Z. Jiexian, L. Xiupeng, and F. Yu, "Multiscale distance coherence vector algorithm for content-based image retrieval," *The Scientific World Journal*, vol. 2014, Article ID 615973, 13 pages, 2014.

[44] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[45] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings Of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, UK, September 1988.

[46] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1, pp. 525–531, 2001.

[47] K. Mikolajczyk and C. Schmid, *An Affine Invariant Interest Point Detector*, pp. 128–142, Springer, Berlin, Germany, 2002.

[48] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Lecture Notes in Computer Science*, vol. 3951, pp. 404–417, Springer, Berlin, Germany, 2006.

[49] J. Shi and Tomasi, "Good features to track," in *Proceedings Of IEEE Conference On Computer Vision And Pattern Recognition CVPR-94*, pp. 593–600, Seattle, WA, USA, June 1994.

[50] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[51] E. Rosten and T. Drummond, *Machine Learning For High-Speed Corner Detection*, Springer, Berlin, Germany, pp. 430–443, 2006.

[52] S. M. Smith and J. M. Brady, "SUSAN---A new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[53] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[54] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings Of the International Conference On Computer Vision-Volume 2*, p. 1150, Norwich, UK, September 1999.

[55] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2911–2918, Washington, DC, USA, June 2012.

[56] G. Yu and J.-M. Morel, "ASIFT: an algorithm for fully affine invariant comparison," *Image Processing On Line*, vol. 1, 2011.

[57] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[58] S. Zhang, Q. Tian, K. Lu, Q. Huang, and W. Gao, "Edge-SIFT: discriminative binary descriptor for scalable partial-duplicate mobile search," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 22, no. 7, pp. 2889–2902, 2013.

[59] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: a SIFT descriptor with color invariant characteristics," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 2 (CVPR'06)*, vol. 2, pp. 1978–1983, New York, NY, USA, June 2006.

[60] W. Cheung and G. Hamarneh, "NSIFT: N-DIMENSIONAL scale invariant feature transform for matching medical images," in *Proceedings of the 2007 4th IEEE International Symposium on Biomedical Imaging: from Nano to Macro*, pp. 720–723, Arlington, Virginia, April 2007.

[61] K. Yan and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 506–513, Washington, DC, USA, June 2004.

[62] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proceedings Of the 11th European Conference On Computer Vision: Part IV*, pp. 778–792, Heraklion, Greece, September 2010.

[63] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Binary Robust invariant scalable keypoints," in *Proceedings Of the IEEE International Conference On Computer Vision*, pp. 2548–2555, Barcelona, Spain, November 2011.

[64] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *Proceedings Of the 2011 International Conference On Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.

[65] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: fast retina keypoint," in *Proceedings Of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 510–517, Providence, RI, USA, June 2012.

[66] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: a comprehensive survey," in *Proceedings of the 18th International Conference on Industrial Technology (ICIT)*, Ontario, Canada, March 2017.

[67] V. P. Singh and R. Srivastava, "Improved image retrieval using color-invariant moments," in *Proceedings of the 2017 3rd International Conference On Computational Intelligence & Communication Technology (CICT)*, pp. 1–6, Ghaziabad, India, February 2017.

[68] A. K. Alhassan and A. A. Alfaki, "Color and texture fusion-based method for content-based image retrieval," in *Proceedings of the 2017 International Conference On Communication, Control, Computing And Electronics Engineering (ICCCCEE)*, pp. 1–6, Khartoum, Sudan, January 2017.

[69] D. R. Dhotre and G. R. Bamnote, "Multilevel haar wavelet transform and histogram usage in content based image retrieval system," in *Proceedings of the 2017 International*

Conference On Vision, Image And Signal Processing (ICVISP), pp. 82–87, Osaka, Japan, September 2017.

[70] J.-Z. Hua, G.-H. Liu, and S.-X. Song, "Content-based image retrieval using color volume histograms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 11, 2019.

[71] M. Alkhawlani, M. Elmogy, and H. Elbakry, "Content-based image retrieval using local features descriptors and bag-of-visual words," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9, 2015.

[72] M. K. Alsmadi, "An efficient similarity measure for content based image retrieval using memetic algorithm," *Egyptian Journal of Basic and Applied Sciences*, vol. 4, no. 2, pp. 112–122, 2017.

[73] V. P. Singh and R. Srivastava, "Effective image retrieval based on hybrid features with weighted similarity measure and query image classification," *International Journal of Computational Vision and Robotics*, vol. 8, no. 2, p. 98, 2018.

[74] S. Jabeen, Z. Mehmood, T. Mahmood, T. Saba, A. Rehman, and M. T. Mahmood, "An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model," *PLoS One*, vol. 13, no. 4, Article ID e0194526, 2018.

[75] Z. Mehmood, F. Abbas, T. Mahmood, M. A. Javid, A. Rehman, and T. Nawaz, "Content-based image retrieval based on visual words fusion versus features fusion of local and global features," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7265–7284, 2018.

[76] Z. Mehmood, T. Mahmood, and M. A. Javid, "Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine," *Applied Intelligence*, vol. 48, no. 1, pp. 166–181, 2018.

[77] P. MANIPOONCHELVI and K. MUNEESWARAN, "Multi region based image retrieval system," *Sadhana*, vol. 39, no. 2, pp. 333–344, 2014.

[78] J. Mukherjee, J. Mukhopadhyay, and P. Mitra, "A survey on image retrieval performance of different bag of visual words indexing techniques," in *Proceedings Of the 2014 IEEE Students' Technology Symposium*, pp. 99–104, Kharagpur, India, March 2014.

[79] P. Sundara Vadivel, D. Yuvaraj, S. Navaneetha Krishnan, and S. R. Mathusudhanan, "An efficient CBIR system based on color histogram, edge, and texture features," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 12, 2019.

[80] A. Du, L. Wang, and J. Qin, "Image retrieval based on colour and improved NMI texture features," *Automatika*, vol. 60, no. 4, pp. 491–499, 2019.

[81] C. Jin and S.-W. Ke, "Content-based image retrieval based on shape similarity calculation," *3D Research*, vol. 8, no. 3, p. 23, 2017.

[82] A. M. Ahmed, S. M. Saadi, and K. N. Hussein, "Image retrieval based on chain code algorithm using color and texture features," *Journal of Kufa for Mathematics and Computer*, vol. 4, no. 2, pp. 18–26, 2017.

[83] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[84] S. A. Nene, S. K. Nayar, and H. Murase, *Columbia Object Image Library (COIL-20)*, Columbia University, New York, NY, USA, 1996.

[85] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 2 (CVPR'06)*, pp. 2161–2168, New York, NY, USA, June 2006.

[86] V. R. Chandrasekhar, "The stanford mobile visual search data set," in *Proceedings of the second annual ACM conference on Multimedia systems-MMSys '11*, p. 117, San Jose, CA, USA, February 2011.

[87] H. Jegou, M. Douze, and C. Schmid, *Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search*, pp. 304–317, Lok Jagruti Kendra, Ahmedabad, India, 2008.

[88] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MI, USA, June 2007.

[89] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: improving particular object retrieval in large scale image databases," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Anchorage, AK, USA, June 2008.

[90] ETH, "ETHZ ZuBud-computer vision lab: Zurich building image database," 2003, http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html.

[91] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, "Scalable triangulation-based logo recognition," in *Proceedings Of ACM International Conference On Multimedia Retrieval (ICMR 2011)*, Trento, Italy, April 2011.

[92] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol, "Scalable logo recognition in real-world images," in *Proceedings Of the 1st ACM International Conference On Multimedia Retrieval-ICMR '11*, pp. 1–8, Trento, Italy, April 2011.

[93] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proceeding Of the 1st ACM International Conference on Multimedia Information Retrieval-MIR '08*, p. 39, Vancouver, Canada, October 2008.

[94] S. Wang and S. Jiang, "INSTRE," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 3, pp. 1–21, 2015.

[95] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[96] P. Griffin, G. Holub, AD. Perona, G. Griffin, A. Holub, and P. Perona, "The Caltech 256, Caltech mimeo," 2007, http://authors.library.caltech.edu/7694%5Cnhttp://www.vision.caltech.edu/Image_Datasets/Caltech256/.

[97] J. Z. Wang, L. Jia, and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[98] Li Jia and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003.

[99] D. Zhang and G. Lu, "Evaluation of similarity measurement for image retrieval," in *Proceedings Of The International Conference On Neural Networks And Signal Processing*, Nanjing, China, December 2003.

[100] Y. Shikhar and V. Prakash Singh, "Comparative analysis of distance metrics for designing an effective content-based image retrieval system using colour and texture features," *International Journal of Image, Graphics and Signal Processing*, vol. 9, no. 12, pp. 58–65, 2017.

[101] D. Srivastava, K. N. Plataniotiss, and A. N. Venetsanopoulos, "Distance measures for color image retrieval," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, pp. 770–774, Chicago, IL, USA, June 1998.

[102] S. D. Thepade and N. B. Yadav, "Assessment of similarity measurement criteria in Thepade's sorted ternary block truncation coding (TSTBTC) for content based video retrieval," in *Proceedings of the 2015 International Conference On Communication, Information & Computing Technology (ICCICT)*, pp. 1–6, Mumbai, India, January 2015.