Journal Pre-proof

A novel image retrieval method based on multi-features fusion

Dongmei Niu, Xiuyang Zhao, Xue Lin, Caiming Zhang

PII: S0923-5965(20)30110-7
DOI: https://doi.org/10.1016/j.image.2020.115911
Reference: IMAGE 115911

To appear in: *Signal Processing: Image Communication*

Received date : 21 September 2019
Revised date : 8 June 2020
Accepted date : 8 June 2020

Please cite this article as: D. Niu, X. Zhao, X. Lin et al., A novel image retrieval method based on multi-features fusion, *Signal Processing: Image Communication* (2020), doi: https://doi.org/10.1016/j.image.2020.115911.

- Micro-structure descriptor is modified to capture more features of an image.
- A new image descriptor is proposed to capture fusion features of an image.
- Similarity between two images is updated with a weighted adjacent structure.
- The proposed method obtains superior performance in content-based image retrieval.

A novel image retrieval method based on multi-features fusion

Dongmei Niu[a], Xiuyang Zhao[a,*], Xue Lin[a], Caiming Zhang[b]

[a]*Shandong Provincial Key Laboratory of Network based Intelligent Computing, University of Jinan, Jinan 250022, China*
[b]*School of Computer Science and Technology, Shandong University, 1500 Shunhua Road, Jinan 250101, China*

Abstract

Image retrieval is an important research topic in pattern recognition and computer vision. We propose a novel content-based image retrieval method. At the image description stage, our method first modifies the traditional microstructure descriptor (MSD) to capture the direct relationship between the shape and texture features and that between the color and texture features. Our method then extracts the uniform local binary patterns (LBP) histogram of the image to capture the color difference information. At the image comparison stage, our method first compares the descriptors of the images to compute their similarities. The similarity between each pair of images is then updated by also considering the similarities to comparable images within the dataset. Accordingly, our method obtains the final similarities of the images. Experimental results demonstrate the effectiveness of our method for image retrieval.

Key words: image retrieval, image description, image comparison, modified MSD, uniform LBP histogram, weighted adjacent structure

1. Introduction

The development of digital image processing technology has increased the number of generated images. Retrieving images that are similar to the given

*Corresponding author
Email addresses: dniu_ujn@hotmail.com (Dongmei Niu), xiuyangzhao@hotmail.com (Xiuyang Zhao), ise_linx@ujn.edu.cn (Xue Lin), czhang@sdu.edu.cn (Caiming Zhang)

query image from a large image dataset becomes a fundamental and important research topic in various areas, such as image processing, pattern recognition, and computer vision. Existing image retrieval methods can generally be divided into three main categories: text-based image retrieval (TBIR), content-based image retrieval (CBIR), and semantic-based image retrieval (SBIR) methods [1–3]. The TBIR methods retrieve images on the basis of image annotations. The performance of such methods is generally low because the manually made annotations are subjective. Annotating images for a huge image dataset also requires much time and human labor. The CBIR methods retrieve images on the basis of low-level features, such as color, texture, shape, or spatial layout [2]. Such methods have higher discriminative power than TBIR ones. However, low-level features cannot describe high-level semantics, and human use high-level semantics to distinguish different images. The SBIR methods are explored to reduce the semantic gap between the low-level features and the high-level semantics. However, these methods remain challenging due to the limitations of current artificial intelligence and related techniques [1, 2, 4]. To date, the most commonly used approach is CBIR. Improving the retrieval accuracy of the CBIR methods is worth exploring. Thus, we focus on this kind of methods in this study.

Using one low-level feature to distinguish different images usually has some limitations. To overcome the limitations, many methods based on fusing several low-level features are proposed. The image retrieval method based on micro-structure descriptor (MSD) [1] is a well-known method that represents an image based on micro-structures. The micro-structures are defined on the basis of micro-structure maps and underlying colors. Such maps are computed by considering the direct relationship between the shape and texture features. The underlying colors refer to the colors under the micro-structure maps. The features extracted on the basis of the micro-structures are used to represent an image. The MSD method can capture the valuable features of an image effectively. However, it disregards the direct relationship between the color and texture features and cannot capture the local color difference well. Therefore,

the retrieval accuracy of this method can be improved.

To improve the performance of the MSD approach [1], we propose a new CBIR method. Color, texture, and shape features are the three low-level features commonly used for image retrieval. The micro-structure map defined in the MSD method [1] captures the direct relationship between the shape and texture features of an image. The direct relationship between color and texture features can also reflect certain characteristics of an image. Therefore, our method modifies the computation of the micro-structure map by considering the relationship between the shape and texture features and the relationship between the color and texture features. In this way, more characteristics of an image are captured. Similar objects in two similar images may have different colors. Thus, comparing the images by differentiating only their color features may falsely result in a large dissimilarity. To decrease the possibility of falsely computing such a large dissimilarity, our method uses the uniform local binary patterns (LBP) histograms to capture local color difference information of the images. To compare the images within the dataset $\boldsymbol{I}$, our method first computes the dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ of the images by comparing their modified MSD (MMSD) features and uniform LBP histograms, respectively. The method then merges $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ and obtains a similarity matrix $\boldsymbol{S}$ of the images. Noise within similar images may also falsely result in a large dissimilarity. We let $N^k_{I_X} \subset \boldsymbol{I}$ be the top k most similar images of the image $I_X \in \boldsymbol{I}$. If images $I_A \in \boldsymbol{I}$ and $I_B \in \boldsymbol{I}$ are similar, then those images within $N^k_{I_A}$ and $N^k_{I_B}$ are somewhat similar to I_B and I_A, respectively. The images within $N^k_{I_A}$ and $N^k_{I_B}$ also present similarities. Therefore, we update the similarity between I_A and I_B by considering the weighted similarities between I_A and I_B, $N^k_{I_A}$ and I_B, $N^k_{I_B}$ and I_A, and $N^k_{I_A}$ and $N^k_{I_B}$ to reduce the noise influence on the retrieval accuracy. In this way, the similarity matrix $\boldsymbol{S}$ is updated to a final similarity matrix $\boldsymbol{S'}$. With $\boldsymbol{S'}$, our method obtains the retrieved images, that is, the sorted similar images, for each image within the dataset. We demonstrate the effectiveness of our method by comparing it with some state-of-the-art methods on five commonly used datasets, namely, Corel-1K [5], Corel-5K [1], Corel-10K

[1], GHIM-10K [4], and CIFAR-10 [6] datasets.

The main contributions of this study are as follows: i) an image description method that can capture the texture, shape, color and color layout features of the images; and ii) a novel image retrieval method that distinguishes two images within a dataset by considering the similarity between these images and their similarities to comparable images within the dataset. The retrieval accuracy of the proposed method is improved compared with those of the state-of-the-art image retrieval methods.

The rest of this paper is organized as follows: Section 2 summarizes the related work on image retrieval. Section 3 provides an overview of the proposed image retrieval method. Sections 4 and 5 discuss the details of the two main stages of the proposed method, namely, extracting the features of the images and comparing the images. Section 6 presents the experimental results generated on five commonly used datasets, and Section 7 concludes the findings.

2. Related work

Many methods have been proposed for CBIR. These methods are based mainly on two types of visual features: local and global features [2, 7]. Local feature-based methods capture low-level features from the key points or salient patches of an image, while global feature-based methods consider the whole image as a salient region and apply convolution on it [7].

Many local feature-based methods for image description and retrieval are available. A common local feature-based method is the scale-invariant feature transform (SIFT) method [8], which extracts a 128-dimensional feature vector for each key point. The feature vector is invariant to image scale and rotation and is robust to a substantial range of affine distortion, addition of noise, and certain levels of illumination changes. However, an image can have many key points, and this condition results in a high-dimensional descriptor for large-scale image retrieval. To reduce computational complexity, Bay et al. [9] proposed a a compact key point descriptor named speeded up robust features (SURF) as an

efficient alternative. For each key point, SURF is a 64-dimensional descriptor. However, this descriptor performs poorly in terms of rotational invariance [7]. The descriptor based on histograms of oriented gradient (HOG) [10] is used mainly for object recognition tasks. This descriptor is invariant to illumination and shadowing effects but has high computational complexity. The methods based on bag-of-visual words (BOW) or its variants borrow techniques from text retrieval, and they are typically used in object-based image retrieval and scene categorization [11–17]. The BOW method proposed in [11] maps the local features (e.g., SIFT, SURF and HOG) of an image into a set of visual words and represents the image as a histogram of visual word occurrences. This method has high computational complexity because the visual words usually come from clustering implementation. Two major limitations of the visual words are lack of semantic information and ambiguity of visual words [4]. Some local-pattern-based methods have been proposed to represent images, and they are used mainly for texture classification and analysis. Ojala et al. [18] presented LBP to describe an image. For each local region of the image, the LBP method thresholds the gray values of the center pixel's neighboring pixels into a binary pattern by comparing the neighboring pixels' and center pixel's gray values. The histogram of the binary patterns can be used as a texture descriptor for the image. However, the LBP method is not rotation invariant. The LBP method has been further extended to the uniform LBP method [19], which represents an image as a histogram of 58 uniform patterns and nonuniform pattern occurrences. The uniform LBP method is rotation invariant. Heikkilä et al. [20] combined the strengths of SIFT and LBP and proposed a texture feature called center-symmetric LBP (CS-LBP). The CS-LBP descriptor is not developed for image retrieval but for object recognition [1]. A local feature descriptor named local neighborhood difference pattern [21] has been proposed to describe an image by transforming the mutual relationship of all neighboring pixels in a binary pattern.

Many global feature-based methods have been proposed for CBIR. These methods describe an image on the basis of low-level information, such as color,

texture, shape, and spatial information [7].

Color belongs to the wavelength-dependent perception [22] that is widely used for image retrieval and object recognition. Swain and Ballard [23] proposed the concept of color histogram (CH) that is invariant to the orientation and scale of an image. However, CH is sensitive to noisy interferences, such as lighting intensity changes and quantization errors. CH also ignores any spatial information [24]. Instead of assigning each pixel of an image into one of the histogram bins only, Han and Ma [24] proposed fuzzy CH (FCH) that considers the color similarity of each pixel's color associated to all the histogram bins via fuzzy-set membership function. The FCH method is less sensitive to inference and more efficient than the CH method. However, spatial information remains missing. Pass et al. [25] defined a color coherence vector (CCV) that incorporates spatial information to describe an image. Each pixel within each histogram bin is classified into either coherent or incoherent depending on whether it belongs to a large similarity-colored region. The CCV consists of the number of coherent versus incoherent pixels with each color. Huang et al. [26] proposed an image descriptor called color correlogram, which characterizes the color distributions of pixels and the spatial correlation of colors. Wang and Chen [27] proposed a fast fractal encoding method to extract image features, and it is based on a non-searching and adaptive quadtree division. In the MPEG-7 standard [28], various color descriptors are proposed. These descriptors include a histogram descriptor that is coded using the Haar transform, a color layout descriptor, a dominant color descriptor and a color structure histogram. Zeng et al. [29] added the spatial information to CH and proposed an image descriptor that represents an image as a spatiogram of colors quantized by Gaussian Mixture Model. The Gaussian Mixture Model was learned by the Expectation-Maximization (EM) algorithm from the training images, and this descriptor is usually called SoC-GMM [7]. Subrahmanyam et al. [30] modified the motif co-occurrence matrix (MCM) method [31] by considering the interrelationship between RGB color planes, and this method is called modified color MCM (MCMCM). Wang et al. [32] proposed quaternion polar harmonic Fourier

moments to capture the relationship between color components within a specific color space. This method can be further used for image retrieval.

Texture is also a primary visual feature of an image, and some texture-based descriptors have been proposed. Haralick et al. [33] extracted the texture feature of an image based on statistics, and this feature is usually called gray-level co-occurrence matrix (GLCM). Wang et al. [34] used the color co-occurrence matrix to extract the texture feature and measure the similarity of two color images. Cross and Jain [35] explored the use of Markov random fields as texture models. Reference [36] proposed the use of Gabor wavelet features for texture analysis. In the MPEG-7 standard [28], three texture descriptors, namely, homogeneous texture, texture browsing, and edge histogram descriptors, are adopted. Compared with the color feature, the texture feature is more expressive of high-level semantic concepts because it can represent internal spatial structure information of an image [7].

In addition to the color and texture features, shape feature is another a primary visual feature because human can identify objects solely on the basis of their shapes. Reference [37] applied the Hahn moments to extract the local and global features of an image. Mahmoudi et al. [38] proposed a feature vector for shape-based image retrieval that classifies image edges on the basis of the edges' orientations and the correlation between neighboring edges. Wang et al. [39] proposed ternary radial harmonic Fourier moments to deal with stereo images in a holistic manner by using ternary number theory and radial harmonic Fourier moments as basis. The proposed moments can be further applied to stereo image retrieval. Kunttu et al. [40] put forward a new Fourier-based descriptor for image retrieval, and it combines the benefits of the wavelet and Fourier transforms. In the MPEG-7 standard [28], three shape descriptors, namely, 3D shape descriptor, region-based shape derived from Zernike moments, and curvature scale space (CSS) descriptor, are used. The shape feature extraction is limited in practice because it requires image segmentation in certain cases [2].

Since image descriptors based on extracting a simple feature usually have some limitations, many descriptors have been proposed on the basis of multi-

features fusion. Liu and Yang [2] proposed an image descriptor called color difference histogram (CDH), which combines the edge orientation, color, and perceptually uniform color difference in $L*a*b*$ space. To express color and texture information equivalently. Feng et al. [41] presented an image descriptor called global correlation descriptor (GCD) that extracts color and texture features. Dubey et al. [42] fused the color and texture features together to construct a hybrid image descriptor. Liu and Yang [43] proposed an image descriptor based on texton co-occurrence matrix (TCM). This descriptor can express the spatial correlation of textons by calculating statistical information of the textons in RGB color image. The TCM utilizes entropy, energy, contrast and homogeneity to describe an image, but these four features are unsuitable to effectively describe an image [7]. Liu et al. [44] further improved the TCM to a new image descriptor called multi-texton histogram (MTH). MTH integrates the advantages of co-occurrence matrix and histogram and can capture the spatial correlation of color and texture orientation. The MSD [1] is defined by extracting the spatial correlations of the underlying colors in micro-structures. The micro-structures are defined on the basis of edge orientation similarity. The MSD integrates color, texture, shape, and color layout information as a whole to describe an image. Liu et al. [4] further proposed an image descriptor called saliency structure histogram (SSH) to stimulate an orientation-selective mechanism for image representation. The SSH can capture the color, edge orientation, and intensity information of an image. The structure elements' descriptor (SED) [3] is constructed to describe an image by extracting the spatial correlation of color and texture information. Reference [45] proposed multi-factors correlation (MFC) to describe an image through structure element correlation, gradient value correlation, and gradient direction correlation. Unar et al. [46] proposed an effective image retrieval strategy based on low-level visual salient features and color segmentation. Zhang et al. [47] combined the mutual information and self-information descriptors and proposed the hybrid information descriptor (HID). The HID can capture the internal correlations of different image feature spaces with image structure and multi-scale analysis [47]. Raza et

al. [7] put forward an image descriptor called correlated primary visual texton histogram features (CPV-THF) by integrating the visual content and semantic information of an image. This descriptor can capture the correlations among the texture, color, intensity, and local spatial structure information of an image. The square texton histogram [48] is constructed on the basis of the correlation between color and texture information extracted by using a four-directional co-occurrence matrix. Singh and Kaur [49] combined the block difference of inverse probabilities (BDIP), block variation of local correlation coefficients (BVLC), and CH to represent an image. Zhou et al. [50] proposed an image descriptor based on fusing CH and local directional pattern. This descriptor was further improved by also considering the SIFT feature of an image [51]. Bella and Vasuki [52] proposed the Fused Information Feature-based Image Retrieval System (FIF-IRS), which is composed of 8-Directional Gray Level Co-occurrence Matrix (8D-GLCM) and HSV Color Moments (HSVCM). Dawood et al. [53] proposed a feature descriptor, Correlated Microstructure Descriptor (CMSD), by correlating color, texture orientation, and intensity information. Wei and Liu [54] also explored the color, edges, and intensity information of images to describe an image.

Textual images are a common type of images. Some methods have been proposed to extract visual features and social tags given that the appearance of text within images is certainly a rich information for humans. Reference [55] detected the text in an image and exploited it as keywords and tags for automatic textual image retrieval. Unar et al. [56] proposed a method that exploits visual and textual information and fuses the two using a Kernel-based method to retrieve the similar images. Reference [57] further used bag of textual words and BOW model to store the textual and visual features, respectively, and combined these features to represent an image.

Many learning-based methods have been proposed in recent years with the development of machine learning and deep learning. Irtaza et al. [58] proposed an image retrieval method based on the concept of semantic class association through trained neural networks. This method fuses wavelet packets and eigen-

values of Gabor filters to represent images. Partial supervised learning scheme based on K-nearest neighbors of a query image is introduced to ensure semantically correct image retrieval. Before retrieving similar images for the query, a few methods [59–62] first narrow down the retrieving space by using a certain classifier to identify the class label of the query image. Consequently, the similarity measure is only performed between the query image and the images in the specific class. Finally, similar images are ranked by their similarities and returned as the retrieval results.

3. Method overview

Image feature extraction and image comparison are the two main components of an image retrieval method. To extract the features of each image within the input image dataset, our method first modifies the original MSD method by directly considering the relationship between the shape and texture features and that between the color and texture features of the image. Our method then computes the uniform LBP histogram of each image to capture color difference information. When comparing the images, our method computes the dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ by comparing the MMSDs and uniform LBP histograms, respectively. Our method merges the normalized $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ and obtains a similarity matrix $\boldsymbol{S}$ between the images. For each image I, we let N_I^k be the set of the first k similar images computed according to $\boldsymbol{S}$. For each pair of images I_A and I_B, our method updates their similarity on the basis of a weighted adjacent structure (WAS). Specifically, our method updates the similarity between I_A and I_B by considering the similarities of I_A and the images within $N_{I_B}^k$, the similarities of I_B and the images within $N_{I_A}^k$, and the similarities of the images within $N_{I_A}^k$ and $N_{I_B}^k$. By updating the similarities of the images with the above-mentioned operation, we obtain a final similarity matrix $\boldsymbol{S'}$. Considering each image within the dataset as a query image, our method sorts the other images in descending similarities to the query. The image with the largest similarity is the most similar image of the query. Figure 1 shows

the flowchart of our method. The highlighted parts with different colors are the two main components of this method. The details of the two components are discussed in the following two sections.

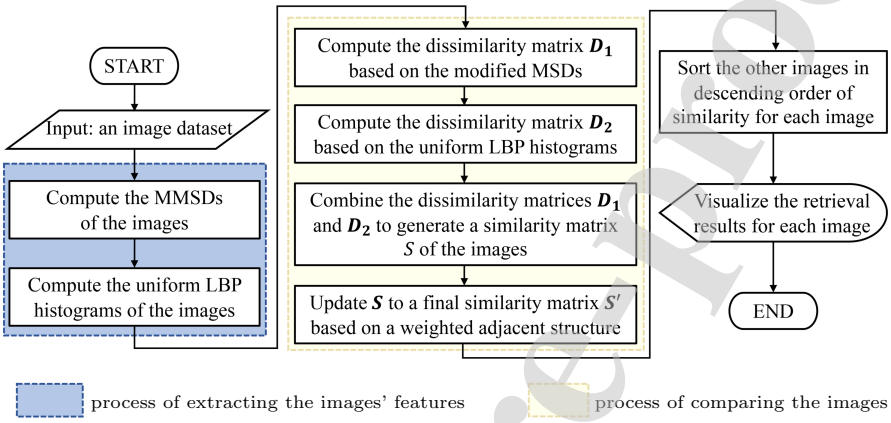


Figure 1: Flowchart of the proposed method.

4. Extracting the features of images

Extracting the features of the images is an important stage of image retrieval. The human visual attention system is directly related to the low-level visual features of images. Thus, we integrate color, texture, shape, and color layout information to represent images. The MSD method [1] describes images with low-level visual features, which computes the edge orientation similarity by considering the relationship between the shape and texture features and uses the underlying colors to form the micro-structure images. The features extracted on the basis of the micro-structure images are used to represent the images. We improve the MSD method by also considering the direct relationship between color and texture features. We also propose an efficient method to compute the MSDs of the images. The similar images may have different colors but similar color difference information. Thus, we further compute the uniform LBP histograms to capture the color difference information of the images.

4.1. Computing the MMSD of an image

Our method improves the original MSD method to compute the MMSD of an image. The HSV color space is more similar to human vision than other color spaces[1]. Thus, we first transform the input RGB image to an HSV image. Following the process of the original MSD method, our method computes the micro-structure map $M(x, y)$ on the basis of the shape and texture features of the image and then computes the micro-structure image $f(x, y)$ on the basis of the micro-structure map and quantized HSV image. To compute the MMSD of the image, our method further calculates another micro-structure map $M'(x, y)$ by considering the relationship between the color and texture features of the image. With $M'(x, y)$ and the quantized HSV image, our method then computes a micro-structure image $f'(x, y)$. With the micro-structure images $f(x, y)$ and $f'(x, y)$ as basis, our method extracts the MMSD for the image. Figure 2 presents the flowchart of the computation of the MMSD. The blue and yellow parts highlight the components of computing the original MSD feature and the improvements for computing the MMSD feature, respectively.

4.1.1. Review of computing the original MSD

The HSV color space can mimic human color perception well. Thus, the input color image is first transformed to an HSV image. The human eye cannot perceive a great number of colors at the same time but can distinguish similar colors well. With this consideration, the MSD method uniformly quantizes the H, S, and V channels into 8, 3, and 3 bins, respectively. The quantizations are

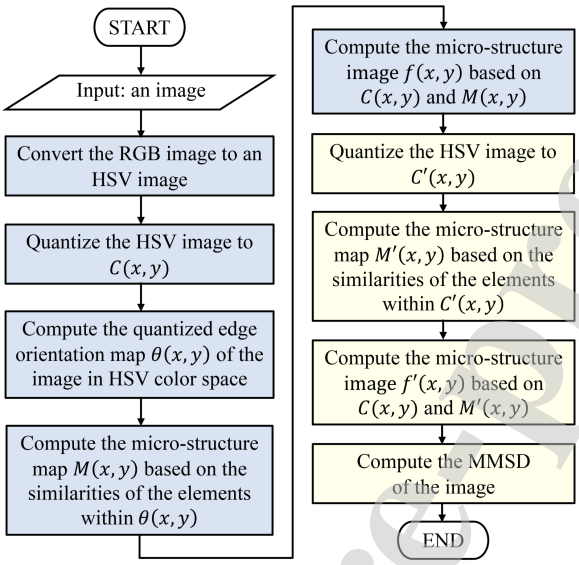


Figure 2: Flowchart of computing the MMSD of an image. The components with blue and yellow backgrounds are those of computing the original MSD of the image and for further computing the MMSD, respectively.

achieved by the following equations:

$$H = \begin{cases} 0, & H \in [0°, 44°] \\ 1, & H \in (44°, 89°] \\ 2, & H \in (89°, 134°] \\ 3, & H \in (134°, 179°] \\ 4, & H \in (179°, 224°] \\ 5, & H \in (224°, 269°] \\ 6, & H \in (269°, 314°] \\ 7, & H \in (314°, 360°], \end{cases} \tag{1}$$

$$S = \begin{cases} 0, & S \in [0, 1/3] \\ 1, & S \in (1/3, 2/3] \\ 2, & S \in (2/3, 1], \end{cases} \tag{2}$$

$$V = \begin{cases} 0, & V \in [0, 1/3] \\ 1, & V \in (1/3, 2/3] \\ 2, & V \in (2/3, 1]. \end{cases} \tag{3}$$

Accordingly, the HSV image is quantized into 72 colors. We use $C(x, y)$ to denote the quantized HSV image. We let $H(x, y), S(x, y)$, and $V(x, y)$ be the three quantized channels of the pixel at (x, y), and the value of $C(x, y)$ is $3 \times 3 \times H(x, y) + 3 \times S(x, y) + V(x, y)$.

After the input image is transformed into an HSV one, the MSD method computes the edge orientation of the image. The orientation map in an image represents the object boundaries, and it can reflect most of the semantic information in the image. For a pixel of the image, its edge orientation is the angle between the gradients along x and y directions. After the edge orientations at the pixels are computed, the MSD method quantizes the orientations into m bins, where $m \in \{6, 12, \ldots, 36\}$. The orientation map of the image can be denoted as $\theta(x, y)$, where $\theta(x, y) = \phi, \phi \in \{0, 1, \ldots, m\}$. The MSD method usually sets the value of m to 6, that is, the orientations are quantized into six bins with an interval of 30°.

The micro-structures in MSD are defined as the collection of colors with similar edge orientation in uniform color space. The MSD method moves a 3×3 block from left to right and top to bottom throughout the image to detect the micro-structures. If each neighbor of the center pixel in a 3×3 block has the same orientation as the center pixel, then the MSD method keeps it and the center pixel; otherwise, the MSD method sets the neighbor to empty. If all the eight neighbors are empty, then the block is not a micro-structure and the center pixel will be set to empty. For the orientation map $\theta(x, y)$ of size $U \times W$, the MSD method moves the 3×3 block from left to right and top to bottom starting from the location $(0, 0)$ with a step-length of three pixels to detect the micro-structures. We let $M_1(x, y)$ be the detected micro-structure map with $0 \leq x \leq U - 1$ and $0 \leq y \leq W - 1$. The micro-structure maps $M_2(x, y)$,

$M_3(x, y)$, and $M_4(x, y)$ can be obtained in the same way starting from the locations $(1, 0)$, $(0, 1)$, and $(1, 1)$, respectively. The MSD method finally merges the four detected micro-structure maps into a micro-structure map $M(x, y)$ and uses $M(x, y)$ as a mask to extract the underlying colors from the quantized HSV image $C(x, y)$. The extracted underlying colors form a micro-structure image $f(x, y)$.

After the micro-structure image $f(x, y)$ is obtained, the value of f at the location (x, y) is $f(x, y) = w, w \in \{0, 1, \ldots, L-1\}$ with L being the number of the quantized colors in $C(x, y)$. For each 3×3 block of $f(x, y)$, we let $P_0(x, y)$ be the center position and $f(P_0) = w_0$. We let $P_i = (x_i, y_i)$ be the eight neighbors of P_0 and $f(P_i) = w_i$, $i = 1, 2, \ldots, 8$. Moving the 3×3 block from left to right and top to bottom throughout $f(x, y)$, the micro-structure features can be defined as follows:

$$H(w_0) = \begin{cases} \dfrac{N\{f(p_0) = w_0 \wedge f(p_i) = w_i| \ |p_i - p_0| = 1\}}{8\bar{N}\{f(p_0) = w_0\}}, \\ where \ w_0 = w_i, \ i \in \{1, 2, \ldots, 8\}. \end{cases} \tag{4}$$

The variable $\bar{N}$ represents the occurring number of w_0, and the variable N denotes the co-occurring number of values w_0 and w_i. The MSD H of the image has a dimensionality of 72.

4.1.2. Computing the MMSD

The MSD method computes the micro-structure map $M(x, y)$ on the basis of the relationship between the shape and texture features and uses the underlying colors to represent an image. We improve the MSD method to compute a micro-structure map $M'(x, y)$ and extract the features of the image based on $M(x, y)$ and $M'(x, y)$. Such a step is conducted to consider the direct relationship between the color and texture features. We also present a method for improving the efficiency of computing the micro-structure maps instead of merging four micro-structure maps $M_1(x, y)$, $M_2(x, y)$, $M_3(x, y)$, and $M_4(x, y)$ to compute the micro-structure map $M(x, y)$ for the image.

Given an image, we compare the feature values of the neighbors on four

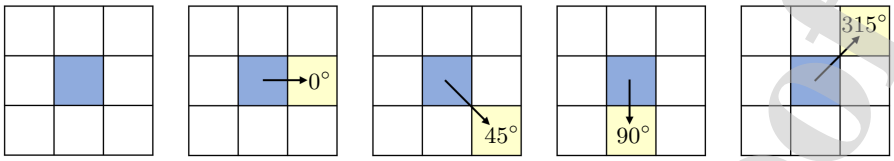


Figure 3: Neighbors within a 3×3 block of a pixel used for computing the micro-structure. The neighbors are on four directions within the 3×3 block of the center pixel shown in (a): (b) 0°, (c) 45°, (d) 90°, and (e) 315°.

directions for each pixel instead of moving a 3×3 block four times to detect the micro-structure map. The center pixel shown in Figure 3(a) is taken as an example. We consider the relationships of the feature values of the center pixel and its four neighbors (see Figure 3(b)-(e)). The corresponding directions of the four neighbors are 0°, 45°, 90° and 315°, respectively. If the feature value of each neighbor is the same as that of the center pixel, then we keep it and the center pixel; otherwise, we set it to empty. For the edge orientation map $\theta(x, y)$ shown in Figure 4(b), Figure 4 shows the extraction of the micro-structure map $M_{0^\circ}(x, y)$ by only considering the neighbor on the 0° direction. Figure 5 shows the merging of micro-structure maps on the four directions (see Figure 5(a)-(d)) to form the final micro-structure map $M(x, y)$ (see Figure 5(e)). The feature value of a center pixel can be compared with those of the neighbors on the four directions at the same time to improve the efficiency of computing the micro-structure map $M(x, y)$. Applying $M(x, y)$ to the quantized color image $C(x, y)$, one can obtain the micro-structure image $f(x, y)$. Figure 6 shows the computation $f(x, y)$.

The micro-structure map $M(x, y)$ reflects the relationship between the shape and texture features of the image. To capture the direct relationship between the color and texture features, we compute a micro-structure map and represent it as $M'(x, y)$. For the input image, we first uniformly quantize the H, S, and V channels into 3, 2, and 2 bins, respectively, to construct an HSV image $C'(x, y)$.

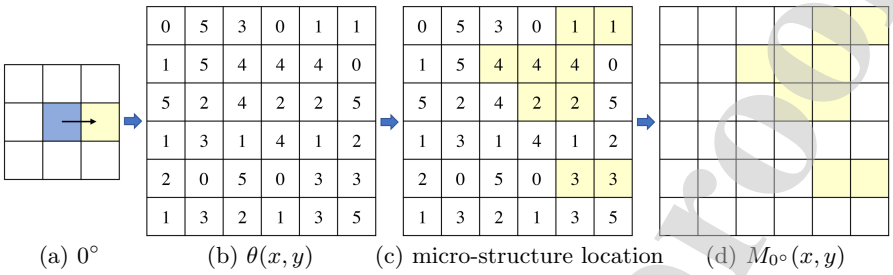


Figure 4: Extraction of the micro-structure map $M_{0^\circ}(x, y)$.

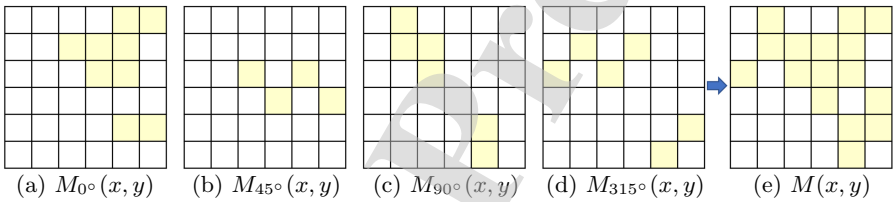


Figure 5: Computation of the fusion of the four maps to form the final micro-structure map $M(x, y)$.

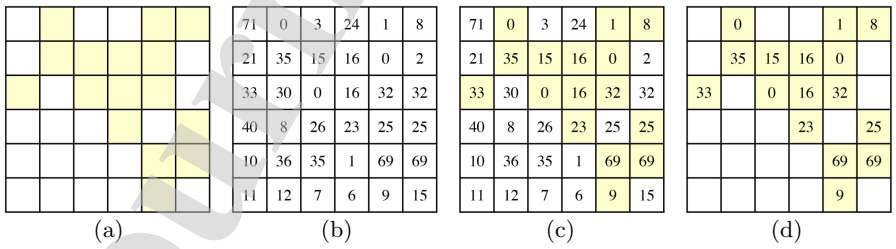


Figure 6: Computation of the micro-structure image $f(x, y)$. (a) $M(x, y)$, (b) quantized color image $C(x, y)$, (c) $M(x, y)$ on $C(x, y)$, (d) micro-structure image $f(x, y)$.

The quantizations are achieved by the following equations:

$$H = \begin{cases} 0, & H \in [0°, 119°] \\ 1, & H \in (119°, 239°] \\ 2, & H \in (239°, 360°], \end{cases} \tag{5}$$

$$S = \begin{cases} 0, & S \in [0, 1/2] \\ 1, & S \in (1/2, 1], \end{cases} \tag{6}$$

$$V = \begin{cases} 0, & V \in [0, 1/2] \\ 1, & V \in (1/2, 1]. \end{cases} \tag{7}$$

Let $H(x, y), S(x, y)$ and $V(x, y)$ be the three quantized channels of the pixel at (x, y), the value of $C'(x, y)$ is $2 \times 2 \times H(x, y) + 2 \times S(x, y) + V(x, y)$. For each pixel of $C'(x, y)$, we compare its value with the values of its neighbors on the 0°, 45°, 90°, and 315° directions to construct a micro-structure map $M'(x, y)$. Applying $M'(x, y)$ to the quantized color image $C(x, y)$, we obtain a micro-structure image $f'(x, y)$. Figure 7 shows the computation of $f'(x, y)$. For the quantized HSV image $C'(x, y)$ shown in Figure 7(a), Figures 7(b)-(e) are the four micro-structure maps detected on different directions. We apply the merged micro-structure map $M'(x, y)$ (see Figure 7(f)) to $C(x, y)$ (see Figure 7(g)). Accordingly, we obtain the micro-structure image $f'(x, y)$ shown in Figure 7(i).

After the micro-structure images $f(x, y)$ and $f'(x, y)$ are obtained, the values of f and f' at location (x, y) are $f(x, y) = w, w \in \{0, 1, \ldots, L-1\}$ and $f'(x, y) = w', w' \in \{0, 1, \ldots, L-1\}$, respectively, where L is the number of the quantized colors in $C(x, y)$. For each 3×3 block of $f(x, y)$ and $f'(x, y)$, we let $f(P_0) = w_0$ and $f'(P_0) = w'_0$ with $P_0(x, y)$ being the center position. We let $P_i = (x_i, y_i)$ be the eight neighbors of P_0 and $f(P_i) = w_i$, $i = 1, 2, \ldots, 8$ and $f'(P_i) = w'_i$, $i = 1, 2, \ldots, 8$. The 3×3 block is moved from left to right and top to bottom throughout $f(x, y)$, and we define the modified micro-structure features

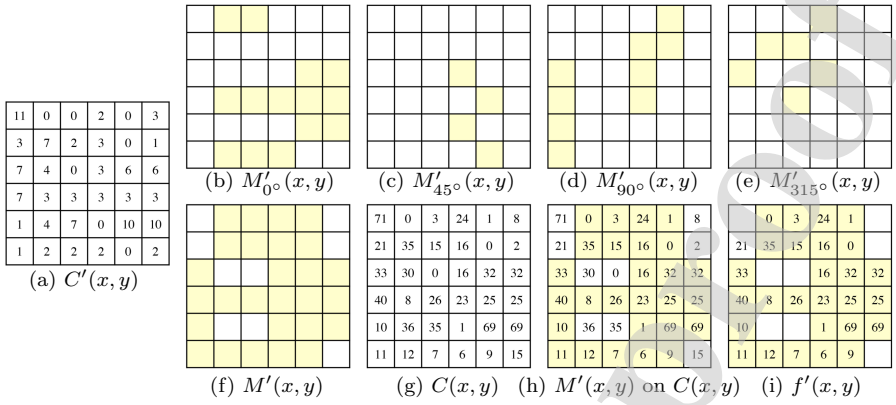


Figure 7: Computation of the micro-structure image $f'(x,y)$.

as follows:

$$MMSD(i) = \begin{cases} H(i) & i \leq L, \\ H'(i-L) & i > L, \end{cases} \tag{8}$$

where $H(i)$ can be computed with Eq. (4) and $H'(i)$ can be computed as

$$H'(w_0) = \begin{cases} \dfrac{N\{f'(p_0) = w'_0 \wedge f'(p_i) = w'_i |\ |p_i - p_0| = 1\}}{8\bar{N}\{f'(p_0) = w'_0\}}, \\ where\ w'_0 = w'_i,\ i \in \{1, 2, \ldots, 8\}. \end{cases} \tag{9}$$

The variable $\bar{N}$ represents the occurring number of w_0, and the variable N denotes the co-occurring number of values w_0 and w_i. The $MMSD$ of the image has a dimensionality of 144. Figure 8 shows the MMSDs of two images selected from the Corel-5K dataset [1].

4.2. Computing the uniform LBP histogram of an image

For each 3×3 block, the micro-structure defined in the MMSD method only considers the similarity of the attributes (e.g., edge orientation and color distribution) of the center pixel and its neighbors. We apply the uniform LBP to compute a histogram for the image for capturing more information about the relationship between the center pixel and its neighbors.

The uniform LBP histogram is originally defined for gray images. It is a histogram with 59 bins. The first 58 bins correspond to the 58 uniform LBP

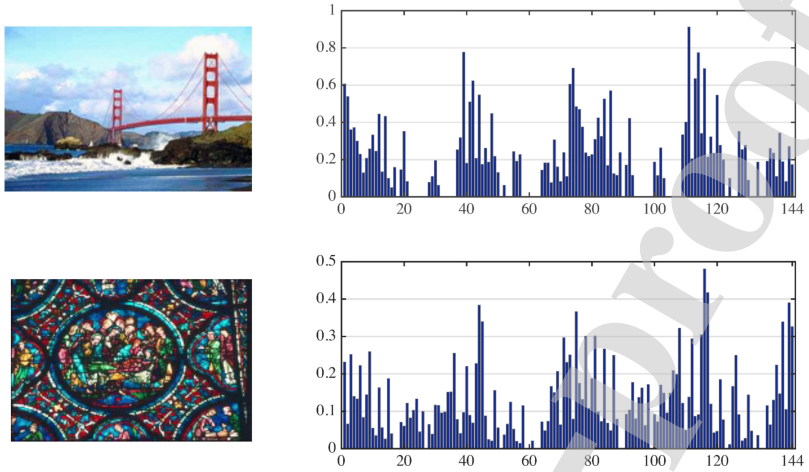


Figure 8: MMSDs of two images selected from the Corel-5K dataset [1].

patterns and the 59th bin corresponds to the nonuniform LBP patterns. To handle color images, we extend a $U \times W$ image to a $U \times W \times 3$ image by extracting its three channels and extract the uniform LBP histogram for the extended image. We let $Lab(x, y)$ be the label of the pixel at position (x, y). The uniform LBP histogram for an extended $U \times W \times 3$ image is as follows:

$$ULBP(i) = \frac{N\{Lab(x, y) = i\}}{U \times W \times 3}, \ i \in [1, 59], \ x \in [1, U], \ and \ y \in [1, 3 \times W]. \tag{10}$$

Figures 9 (a) and (c) show the extended images of those presented in Figure 8. Figures 9 (b) and (d) illustrate the corresponding uniform LBP histograms.

5. Comparing the images

After the features of the images are extracted, our method computes the similarity of the images. To compare the images within a dataset, our method first computes two dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ by comparing the MMSDs and uniform LBP histograms of the images, respectively. Our method then combines the two matrices to generate a similarity matrix $\boldsymbol{S}$ of the images. For each pair of images I_A and I_B, the similarities of I_B and the images that are

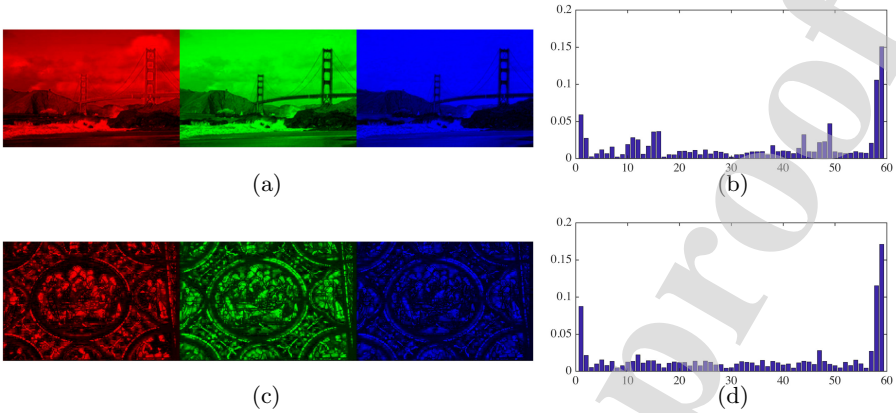


Figure 9: Extended images and the uniform LBP histograms of the two images shown shown in Figure 8.

similar to I_A within the dataset can reflect the similarity between I_A and I_B to an extent. On this basis, our method updates $\boldsymbol{S}$ to a final similarity matrix $\boldsymbol{S'}$ with a WAS. Details are discussed in the remaining part of this section.

5.1. Computing dissimilarity matrix D_1

Given an image dataset, we use $\boldsymbol{D_1}$ to represent the dissimilarity matrix of the images computed by comparing the MMSDs of the images. For a pair of images I_i and I_j within the dataset, we use $MMSD_i$ and $MMSD_j$ to represent their MMSDs. The dissimilarity between I_i and I_j is computed as the L_1 distance between $MMSD_i$ and $MMSD_j$, which is

$$\boldsymbol{D_1}(i,j) = \sum_{k=1}^{144} |MMSD_i(k) - MMSD_j(k)|. \tag{11}$$

5.2. Computing dissimilarity matrix D_2

Similar to the computation of dissimilarity matrix $\boldsymbol{D_1}$, we calculate a dissimilarity matrix $\boldsymbol{D_2}$ of the images by comparing the uniform LBP histograms. For a pair of images I_i and I_j within the database, we use $ULBP_i$ and $ULBP_j$ to represent their LBP histograms. The dissimilarity $\boldsymbol{D_2}(i,j)$ between this pair

of images is the L_1 distance between $ULBP_i$ and $ULBP_j$, which is

$$\boldsymbol{D_2}(i,j) = \sum_{k=1}^{59} |ULBP_i(k) - ULBP_j(k)|. \tag{12}$$

5.3. Computing similarity matrix S

After the dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ are computed, we combine them to generate a similarity matrix $\boldsymbol{S}$ and evaluate the similarities of the images. To combine the two matrices, we first normalize the elements of each matrix to $[0, 1]$ as follows:

$$\boldsymbol{D_1}(i,j) = \frac{\boldsymbol{D_1}(i,j) - min(\boldsymbol{D_1})}{max(\boldsymbol{D_1}) - min(\boldsymbol{D_1})}; \tag{13}$$

$$\boldsymbol{D_2}(i,j) = \frac{\boldsymbol{D_2}(i,j) - min(\boldsymbol{D_2})}{max(\boldsymbol{D_2}) - min(\boldsymbol{D_2})}. \tag{14}$$

The similarity matrix $\boldsymbol{S}$ of the images is computed as:

$$\boldsymbol{S} = 1 - w \times \boldsymbol{D_1} - (1 - w) \times \boldsymbol{D_2}, \tag{15}$$

where the parameter w is the weight of the dissimilarity matrix $\boldsymbol{D_1}$. For all the experimental results shown in this paper, we simply set the value of w to 0.5.

5.4. Updating S to a final similarity matrix S′

For two images I_A and I_B, we use $N^k_{I_A}$ and $N^k_{I_B}$ to represent the top k images within the dataset that are similar to the images I_A and I_B, respectively. If I_A and I_B are similar, then $N^k_{I_A}$ are similar to I_B and $N^k_{I_B}$ are similar to I_A; otherwise, $N^k_{I_A}$ and $N^k_{I_B}$ are not similar to I_B and I_A, respectively. To guarantee the accuracy of the computed similarity $\boldsymbol{S}(I_A, I_B)$ between I_A and I_B, we update $\boldsymbol{S}(I_A, I_B)$ to $\boldsymbol{S'}(I_A, I_B)$ by also considering the relationship between $N^k_{I_A}$ and I_B, $N^k_{I_B}$ and I_A, and $N^k_{I_A}$ and $N^k_{I_B}$. We compute $\boldsymbol{S'}(I_A, I_B)$ as follows:

$$\boldsymbol{S'}(I_A, I_B) = w_1\boldsymbol{S}(I_A, I_B) + w_2 S_N(I_A, N^k_{I_B}) + w_3 S_N(I_B, N^k_{I_A}) + w_4 S_{NN}(N^k_{I_A}, N^k_{I_B}), \tag{16}$$

where w_1, w_2, w_3, and w_4 are the weights, and $w_1 + w_2 + w_3 + w_4 = 1$. We simply set these weights to 1/2, 1/6, 1/6, and 1/6 for all the experimental results

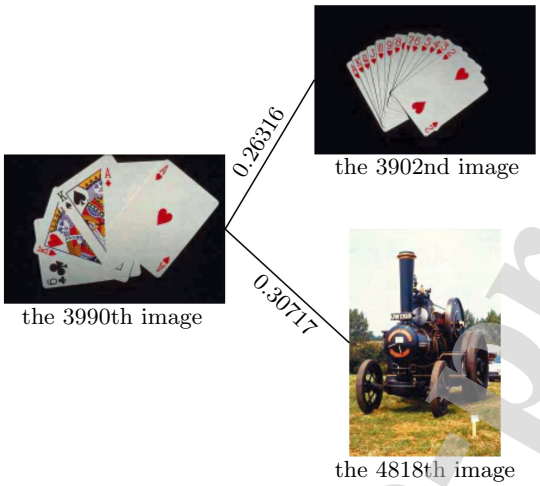


Figure 10: Similarity $\boldsymbol{S}(3990, 3902)$ between the 3990th and 3902nd images and similarity $\boldsymbol{S}(3990, 4818)$ between the 3990th and 4818th images of the Corel-5K dataset [1].

shown in this paper. The variable $S_N(I_A, N^k_{I_B})$ represents the similarity of I_A and $N^k_{I_B}$, and it can be defined as:

$$S_N(I_A, N^k_{I_B}) = \sum_{i=1}^{k} \frac{\boldsymbol{S}(I^i_B, I_B)\boldsymbol{S}(I_A, I^i_B)}{\sum_{j=1}^{k} \boldsymbol{S}(I^j_B, I_B)}, \tag{17}$$

where $I^i_B \in N^k_{I_B}$, $i \in [1, k]$. Similarly, the computation of $S_N(I_B, N^k_{I_A})$ is

$$S_N(I_B, N^k_{I_A}) = \sum_{i=1}^{k} \frac{\boldsymbol{S}(I^i_A, I_A)\boldsymbol{S}(I_B, I^i_A)}{\sum_{j=1}^{k} \boldsymbol{S}(I^j_A, I_A)}. \tag{18}$$

The variable $S_{NN}(N^k_{I_A}, N^k_{I_B})$ denotes the similarity between $N^k_{I_A}$ and $N^k_{I_B}$, which is

$$S_{NN}(N^k_{I_A}, N^k_{I_B}) = \sum_{i=1}^{k}\sum_{j=1}^{k} \frac{\boldsymbol{S}(I^i_A, I_A)\boldsymbol{S}(I^j_B, I_B)\boldsymbol{S}(I^i_A, I^j_B)}{\sum_{u=1}^{k}\sum_{v=1}^{k} \boldsymbol{S}(I^u_A, I_A)\boldsymbol{S}(I^v_B, I_B)}. \tag{19}$$

Figure 10 shows the similarity between the 3990th and 3902nd images and that between the 3990th and 4818th images of the Corel-5K dataset. The values are the similarities. A large value corresponds to a high similarity of the corresponding two images. The 3902nd image is actually more similar to the 3990th image than the 4818th one. However, the computed results show that the 4818th image is more similar to the 3990th one. Figures 11 and 12 show

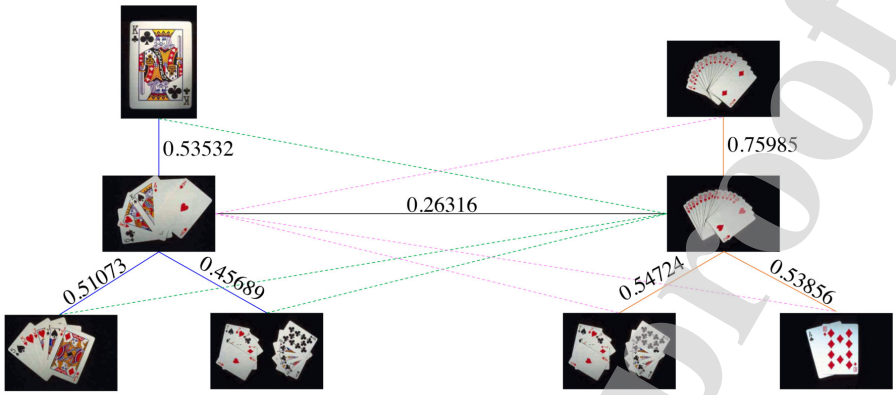


Figure 11: Similarity between the 3990th and the 3902nd images and the similarities of these two images and their respective top three similar images within the Corel-5K dataset [1].

the relationships between these two pairs of images and their corresponding top k ($k = 3$) most similar images, respectively. The similarities between these two pairs of images by updating the similarities based on Eq. (16) are shown in Figure 13. After the similarities are updated, the 3902nd image is more similar to the 3990th image.

Figure 12: Similarity between the 3990th and the 4818th images and the similarities of these two images and their respective top three similar images within the Corel-5K dataset [1].

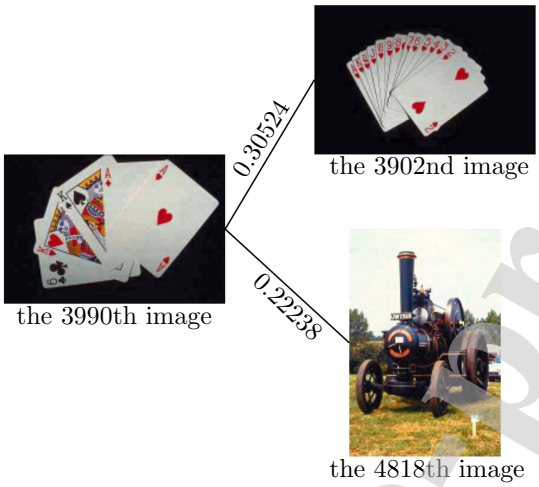


Figure 13: Similarity $\boldsymbol{S'}(3990, 3902)$ between the 3990th and 3902nd images and similarity $\boldsymbol{S'}(3990, 4818)$ between the 3990th and 4818th images of the Corel-5K dataset [1].

6. Results and discussion

We conduct experiments on five commonly used datasets, namely, Corel-1K [5], Corel-5K [1], Corel-10K [1], GHIM-10K [4], and CIFAR-10 [6], to analyze the performance of the proposed method. Concerning the parameter k discussed in Section 5.4, we test setting k to different values and found that our method obtained good retrieval results when k was 3. For all the experimental results shown in this section, the value of k is 3. Tests for determining the value of k are presented in Section 6.3.

6.1. Performance evaluation metrics

Precision and recall are two commonly used measurements to measure the performance of an image retrieval method. For a query image, precision is the fraction of retrieved images that are similar to the query, and recall is the fraction of similar images that are retrieved. The mathematical definitions of precision $P(i, j)$ and recall $R(i, j)$ of the j-th image within the i-th category are as follows:

$$p(i,j) = \frac{NS(i,j)}{NR(i,j)},\ r(i,j) = \frac{NS(i,j)}{ND(i,j)}, \tag{20}$$

where $NS(i,j)$ represents the number of retrieved similar images, $NR(i,j)$ represents the number of retrieved images, and $ND(i,j)$ is the number of similar images in the whole dataset. For an image dataset that includes C categories of images, we use C_i to denote the number of images within the i-th category. The average precision $P(i)$ and recall $R(i)$ for the i-th category are as follows:

$$P(i) = \frac{\sum_{j=1}^{C_i} p(i,j)}{C_i},\ R(i) = \frac{\sum_{j=1}^{C_i} r(i,j)}{C_i}. \tag{21}$$

The average retrieval precision (ARP) and the average retrieval recall (ARR) for the dataset are:

$$ARP = \frac{\sum_{i=1}^{C} P(i)}{C},\ ARR = \frac{\sum_{i=1}^{C} R(i)}{C}. \tag{22}$$

We use the ARP and ARR to evaluate our method and compare our method with some state-of-the-art CBIR methods on the Corel-1K [1], Corel-5K [1], Corel-10K [1] and GHIM-10K [4] datasets.

To evaluate the performance of different methods on the CIFAR-10 dataset [6], we adopt the widely used mean average precision (mAP) criterion. The mAP at top R retrieved images is usually represented as $mAP@R$. The $mAP@R$ is defined as follows:

$$mAP@R = \frac{1}{Q}\sum_{i=1}^{Q} AP(i)@R, \tag{23}$$

where Q is the number of queries and $AP(i)@R$ is the average precision at top R retrieved images of the i-th query. The computation of $AP(i)@R$ is:

$$AP(i)@R = \frac{1}{L}\sum_{r=1}^{R} P_i(r)\delta(r), \tag{24}$$

where L is the number of similar images for the i-th query within the top R retrieved images, $P_i(r)$ represents the precision when top r images are returned for the i-th query, and $\delta(r)$ is an indicator function that indicates 1 when the r-th result is a similar image and 0 otherwise.

6.2. Retrieval results

We present a method for image retrieval by considering the color, texture, shape, and color layout information of the images. Our method first computes a

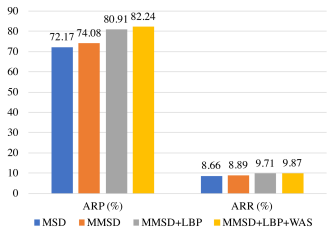


Figure 14: ARPs and ARRs generated by the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-1K dataset with $NR = 12$.

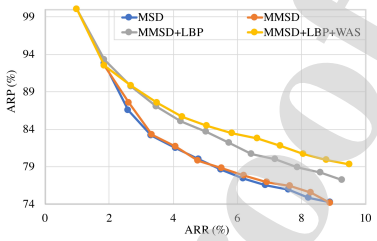


Figure 15: ARP-ARR curves of the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-1K dataset with $NR \in [1, 12]$.

dissimilarity matrix $\boldsymbol{D_1}$ based on computing the MMSDs of the images. Thereafter, the method computes a dissimilarity matrix $\boldsymbol{D_2}$ based on extracting the uniform LBP histograms of the images. Our method then combines $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ to construct a similarity matrix $\boldsymbol{S}$ of the images. Updating $\boldsymbol{S}$ to a final similarity matrix $\boldsymbol{S'}$ with a WAS, our method finally achieves the retrieval results for the images. We use "MMSD" to represent the method of only considering the MMSDs of the images, "MMSD+LBP" to represent the method of combining the MMSDs and the uniform LBP histograms, and "MMSD+LBP+WAS" to present the complete proposed method.

6.2.1. Corel-1K dataset

The Corel-1K dataset [5] contains 1000 images that are classified into 10 different categories. Each category contains 100 images. The 10 categories are Africans, Beaches, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains, and Food. All the images within this dataset are of the same size of either 384×256 or 256×384.

Figure 14 shows the ARPs and ARRs generated by the "MSD" [1], "MMSD", "MMSD+LBP", and "MMSD+LBP+WAS" methods. The value of NR is 12. Figure 15 shows the ARP-ARR curves corresponding to these four methods with NR varying from 1 to 12. The MMSD (i.e., MMSD) achieved better results than the original MSD. The combination of the MMSD and the uni-

Table 1: Comparison of different image retrieval methods on the Corel-1K dataset. The value of NR is 12. The bold values indicate the best results.

Category	Performance	MTH [44]	MSD [1]	CDH [2]	SED [3]	GCD [41]	MCMCM [30]	CPV-THF [7]	OUR
Africans	ARP	69.17	83.33	77.50	82.50	87.50	69.75	**91.66**	86.42
	ARR	8.30	10.00	9.30	9.90	10.50	6.00	**11.00**	9.50
Beaches	ARP	61.67	43.33	56.67	28.33	**68.33**	54.25	54.58	50.25
	ARR	7.40	5.20	6.80	3.40	**8.20**	4.40	6.55	6.03
Buildings	ARP	45.83	63.33	47.50	47.50	61.67	63.95	**78.75**	77.83
	ARR	5.50	7.60	5.70	5.70	7.40	4.30	**9.45**	9.34
Buses	ARP	68.33	76.67	71.67	73.33	80.00	89.65	86.25	**98.08**
	ARR	8.20	9.20	8.60	8.80	9.60	8.00	10.35	**11.77**
Dinosaurs	ARP	**100.00**	**100.00**	**100.00**	90.00	**100.00**	98.70	**100.00**	99.17
	ARR	**12.00**	**12.00**	**12.00**	10.80	**12.00**	11.40	**12.00**	11.90
Elephants	ARP	70.83	65.00	62.50	55.00	67.50	48.80	65.83	**74.50**
	ARR	8.50	7.80	7.50	6.60	8.10	3.35	7.90	**8.94**
Flowers	ARP	75.00	86.67	60.83	72.50	88.33	92.30	95.41	**95.92**
	ARR	9.00	10.40	7.30	8.70	10.60	8.12	11.45	**11.51**
Horses	ARP	100.00	97.50	91.67	62.50	**100.00**	89.45	93.33	95.92
	ARR	12.00	11.70	11.00	7.50	**12.00**	8.40	11.20	11.51
Mountains	ARP	39.17	29.17	44.17	40.00	55.00	47.30	**56.25**	53.25
	ARR	4.70	3.50	5.30	4.80	6.60	3.90	**6.75**	6.39
Food	ARP	52.50	76.67	45.00	64.17	74.17	70.90	85.83	**91.08**
	ARR	6.30	9.20	5.40	7.70	8.90	6.15	10.30	**10.93**
Average	ARP	68.25	72.17	65.75	61.58	72.85	72.50	80.79	**82.24**
	ARR	8.19	8.66	7.89	7.39	9.39	6.40	9.69	**9.87**

form LBP histogram (i.e., MMSD+LBP) performed better than the MMSD method. The ARP and ARR were further improved by applying the WAS (i.e., MMSD+LBP+WAS).

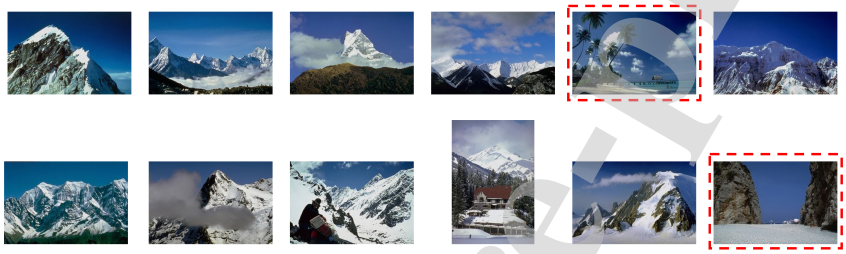
Table 1 lists the ARPs and ARRs generated by different methods on each category of the Corel-1K dataset. We use "OUR" to represent the proposed method, that is, the "MMSD+LBP+WAS" method. The maximum ARP and ARR on each category are highlighted in bold. Compared with the listed methods, our method performed the best on the four categories, namely, "Buses", "Elephants", "Flowers", and "Food". Our method performed the best in terms of the average ARP and ARR over all the 10 categories.

(a) retrieval results of the MSD method [1]

(b) retrieval results of the proposed method

Figure 16: The top 12 similar images for the 88th image of the Corel-1K dataset retrieved by the (a) MSD method [1] and (b) proposed method. The top-left images of (a) and (b) are the same query image. The similar images include the query image itself. The images marked with red dashed boxes are the incorrectly retrieved images.

(a) retrieval results of the MSD method [1]



(b) retrieval results of the proposed method

Figure 17: The top 12 similar images for the 845th image of the Corel-1K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

Figure 16 shows the top 12 retrieved images for the 88th image of the Corel-1K dataset. Figures 16 (a) and (b) present the results generated by the MSD method and the proposed method, respectively. The top-left images of (a) and (b) are the same query image. The retrieved similar images include the query image itself. The images highlighted with red dashed boxes are the incorrectly retrieved ones. For this query image, the precisions of the MSD method and our method are 50% and 100%, respectively. Figure 17 shows the retrieval results of a mountain image within the Corel-1K dataset. The precisions of the MSD method and our method are 66.7% and 91.7%, respectively. In Figure 17(b), the two incorrectly retrieved images belong to the "Beaches" category. Such images have similar color information to those within the "Mountains" category. Therefore, our method performed relatively worse when handling the images within "Beaches" and "Mountains".

We also tried to compare our method with some learning-based methods. Table 2 shows the comparison results on the Corel-1K dataset. The value of

NR is 20. Our method performed better than the first three methods and relatively worse than the remaining three methods. However, our method achieved the highest ARP/ARR on the "Buses", "Horses", and "Food" categories. Our method is easy to implement and is independent of the learning process that takes a large amount of processing time.

Table 2: Comparison of our method with some learning-based methods on the Corel-1K dataset. The value of NR is 20.

Category	Performance	Elalami [59]	Irtaza et al. [58]	Ali et al. [60]	SoC-GMM [29]	Sharif et al. [61]	Vibhav et al. [62]	OUR
Africans	ARP	72.6	65	60.08	72.5	74.19	**84.5**	83.15
	ARR	16.1	13	12.02	14.5	14.83	**-**	16.63
Beaches	ARP	59.3	60	60.39	65.2	75.38	**84.0**	46.20
	ARR	20.3	12	12.08	13.04	**15.07**	-	9.24
Buildings	ARP	58.7	62	69.66	70.6	**75.82**	68.0	72.20
	ARR	19.1	12	13.93	14.12	**15.16**	-	14.44
Buses	ARP	89.1	85	93.65	89.2	81.59	91.0	**97.75**
	ARR	12.6	17	18.73	17.84	16.31	-	**19.55**
Dinosaurs	ARP	99.3	93	99.88	**100**	**100**	90.5	99.05
	ARR	10.9	19	19.98	**20**	**20**	-	19.81
Elephants	ARP	70.2	65	70.76	70.5	**96.70**	66.5	64.30
	ARR	16.3	13	**14.15**	14.1	19.34	-	12.86
Flowers	ARP	92.8	94	88.37	94.8	93.21	**97.5**	95.45
	ARR	12.9	19	17.67	18.96	18.64	**-**	19.09
Horses	ARP	85.6	77	82.77	91.8	85.25	89.5	**92.40**
	ARR	14.4	15	16.55	18.36	17.05	-	**18.48**
Mountains	ARP	56.2	73	61.08	72.25	**80.47**	72.5	47.80
	ARR	23.6	10	12.22	14.45	**16.09**	-	9.56
Food	ARP	77.2	81	65.09	78.8	81.32	82.0	**89.45**
	ARR	14.8	16	13.02	15.76	16.26	-	**17.89**
Average	ARP	76.1	75	75.17	80.57	**84.39**	82.6	78.78
	ARR	16.1	15	15.03	16.11	**16.87**	-	15.76

6.2.2. Corel-5K dataset

The Corel-5K dataset [1] contains 5000 images that are classified into 50 different categories. Each category contains 100 images. The size of each image is either 192×128 or 128×192.

Figure 18 shows the ARPs and ARRs generated by the "MSD" [1], "MMSD", "MMSD+LBP", and "MMSD+LBP+WAS" methods with NR equaling to 12.

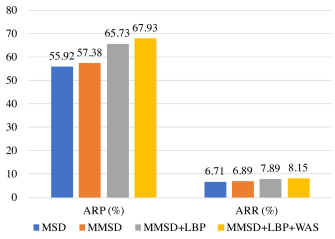


Figure 18: ARPs and ARRs generated by the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-5K dataset with $NR = 12$.

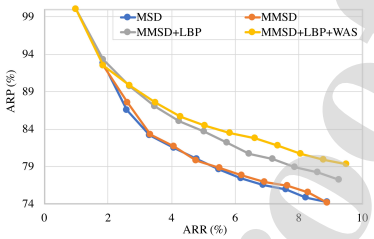


Figure 19: ARP-ARR curves of the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-5K dataset with $NR \in [1, 12]$.

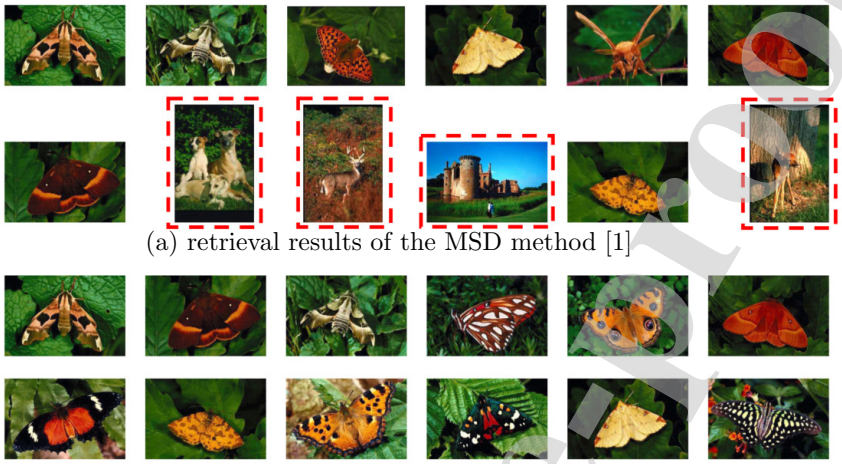
Table 3: Comparison of different image retrieval methods on the Corel-5K dataset. The value of NR is 12. The bold values indicate the best results.

Performance	MTH [44]	MSD [1]	CDH [2]	HID [47]	SoC-G MM [29]	STH [48]	CPV-THF [7]	Sharif et al. [61]	OUR
ARP	49.84	55.92	57.23	51.80	51.80	60.28	63.90	57.3	**67.93**
ARR	5.98	6.71	6.87	6.22	6.22	7.23	7.66	-	**8.15**

Figure 19 shows the ARP-ARR curves corresponding to these four methods with NR varying from 1 to 12. The MMSD method performed better than the MSD method, and the MMSD+LBP method performed better than the MMSD method. The MMSD+LBP+WAS method achieved the best performance.

Table 3 lists the ARPs and ARRs generated by different methods on the Corel-5K dataset. For all the methods listed here, the value of NR is 12. The best ARP and ARR are highlighted in bold. Our method performed the best compared with all the listed methods.

Figures 20 and 21 show the top 12 retrieved images of the 4192nd and 4818th images of the Corel-5K dataset, respectively. The precisions of the MSD method on these two queries are 75% and 66.7%, respectively; while the precisions of our method on these two queries are both 100%. Our method performed better than the MSD method on these two queries.



(a) retrieval results of the MSD method [1]

(b) retrieval results of the proposed method

Figure 20: The top 12 similar images for the 4192nd image of the Corel-5K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

(a) retrieval results of the MSD method [1]

(b) retrieval results of the proposed method

Figure 21: The top 12 similar images for the 4818th image of the Corel-5K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

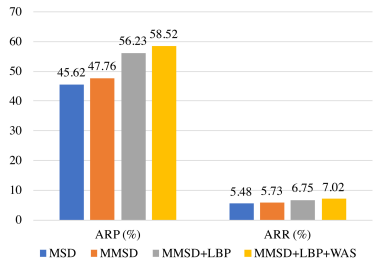


Figure 22: ARPs and ARRs generated by the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-10K dataset with $NR = 12$.

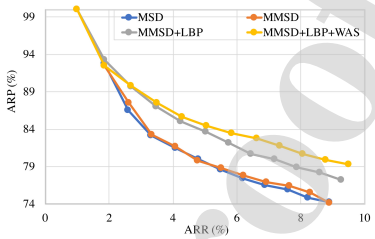


Figure 23: ARP-ARR curves of the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the Corel-10K dataset with $NR \in [1, 12]$.

6.2.3. Corel-10K dataset

The Corel-10K dataset [1] contains 100 categories of images. These 100 categories are composed of all the 50 categories of images of the Corel-5K dataset and another 50 categories of images. Each category contains 100 images. Every image exhibits a size of either 192×128 or 128×192.

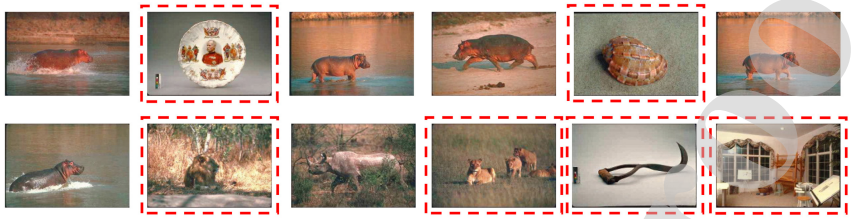
Figure 22 shows the ARPs and ARRs generated by the "MSD" [1], "MMSD", "MMSD+LBP", and "MMSD+LBP+WAS" methods with NR equaling to 12. Figure 23 shows the ARP-ARR curves corresponding to these four methods with NR varying from 1 to 12. The MMSD performed better than the MSD method, and the MMSD+LBP method performed better than the MMSD method. The MMSD+LBP+WAS method performed the best.

Table 4: Comparison of different image retrieval methods on the Corel-10K dataset. The value of NR is 12. The bold values indicate the best results.

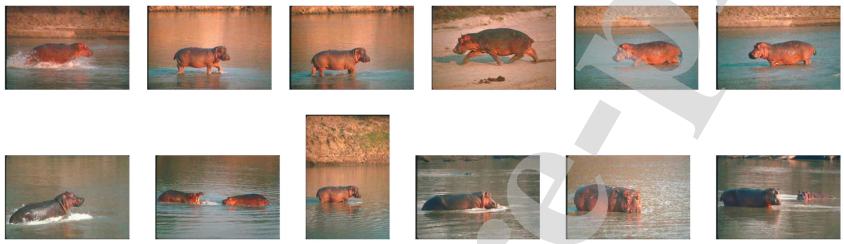
Performance	MTH [44]	MSD [1]	CDH [2]	HID [47]	SSH [4]	SoC-GMM [29]	STH [48]	CPV-THF [7]	OUR
ARP	41.44	45.62	45.24	49.15	54.88	47.25	48.03	52.28	**58.52**
ARR	4.97	5.48	5.43	5.90	6.58	5.67	5.76	6.27	**7.02**

Table 4 lists the ARPs and ARRs generated by different methods on the Corel-10K dataset. For all the methods listed here, the value of NR is 12. The optimal ARP and ARR are highlighted in bold. Our method performed the

best compared with all the listed methods.

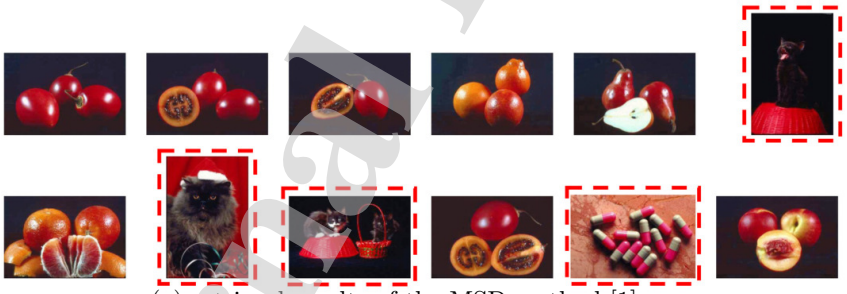


(a) retrieval results of the MSD method [1]

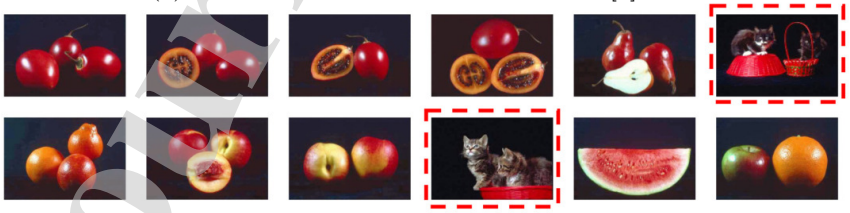


(b) retrieval results of the proposed method

Figure 24: The top 12 similar images for the 4005th image of the Corel-10K dataset retrieved by the (a) MSD method [1] and (b) proposed method.



(a) retrieval results of the MSD method [1]



(b) retrieval results of the proposed method

Figure 25: The top 12 similar images for the 2124th image of the Corel-10K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

Figure 24 shows the top 12 retrieved images of the 4005th image within the Corel-10K dataset. Figures 24 (a) and 24 (b) are the results generated by the MSD method and the proposed method, respectively. The precisions of the MSD method and our method are 50% and 100%, respectively. Figure 25 shows the results of the 2124th image. The precisions of the MSD method and our method are 66.7% and 88.3%, respectively. Our method performed better than the MSD method.

Our method still has limitations despite its relatively good performance. Taking the retrieval results of the 2124th image shown in Figure 25(b) as an example, our method incorrectly considered that the cat image (i.e., the 7448th image of the Corel-10K dataset) shown in the very right of the first row was more similar than the fruit image (i.e., the 2173rd image of Corel-10K dataset) shown in the very left of the second row. The possible reasons for resulting in such negative result are as follows: i) our method combined the color, texture, and shape features to represent an image. For an image, our method extracted the features at each pixel on the basis of the relationship of the pixel and its eight neighboring pixels. Our method constructed the descriptor for the image by counting the number of pixels with similar features. Two dissimilar images may have similar features. The main colors of the query image are black, red, and white. The 7448th image roughly has the similar main colors. The main colors of the 2173rd image are black, yellow, and white. Moreover, the percentages of the 7448th image's three main colors are roughly similar to the percentages of the query image's three corresponding main color. In other words, the color features of the 7448th image are more similar to those of the query image. Thus, our method might mistakenly consider that the 7448th image was more similar to the query one; ii) our method moved a 3×3 block from left to right and top to bottom throughout the image to extract the image descriptor. Specifically, our method extracted the descriptor on the basis of the relationship of each pixel and its eight neighboring pixels. Some pixels of two different images may have similar blocks because a 3×3 block is a small block. Therefore, our method might mistakenly consider that the 7448th image was more similar to

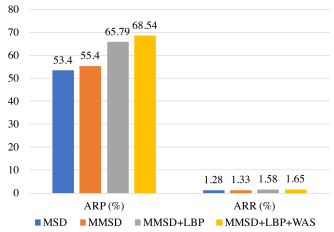


Figure 26: ARPs and ARRs generated by the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the GHIK-10K dataset with $NR = 12$.

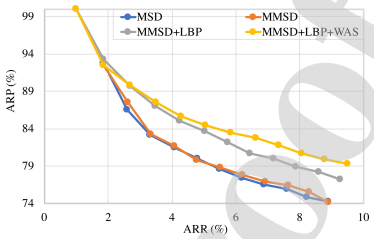


Figure 27: ARP-ARR curves of the MSD, MMSD, MMSD+LBP, and MMSD+LBP+WAS methods on the GHIK-10K dataset with $NR \in [1, 12]$.

the query one. To overcome this type of negative examples, we would like to consider using different sizes of blocks to hierarchically extract the features of the images. For two dissimilar images, some of their pixels may have a similar 3×3 block but dissimilar larger blocks. Small blocks can be used to obtain some detailed features of the images. The larger blocks can obtain relatively global features. We also consider fusing the visual salient features with our proposed image descriptor as one of our future tasks.

6.2.4. GHIM-10K dataset

The GHIM-10 dataset [4] consists of 10,000 images that are classified into 20 different categories. Each category contains 500 images. The size of each image is either 300×400 or 400×300.

Figure 26 shows the ARPs and ARRs generated by the "MSD" [1], "MMSD", "MMSD+LBP", and "MMSD+LBP+WAS" methods with NR equaling to 12. Figure 27 shows the ARP-ARR curves corresponding to these four methods with NR varying from 1 to 12. The MMSD performed better than the MSD method, and the MMSD+LBP method performed better than the MMSD method. The MMSD+LBP+WAS method performed the best.

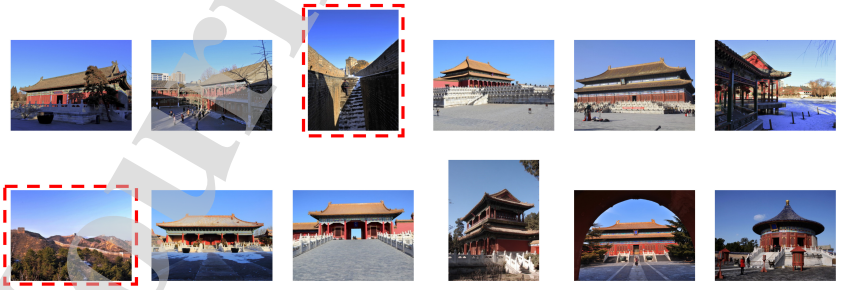
Figures 28 and 29 show the retrieval results of the MSD method [1] and our method on two selected query images within the GHIM-10K dataset. Our method retrieved more similar images for the two queries compared with the

(a) retrieval results of the MSD method [1]

(b) retrieval results of the proposed method

Figure 28: The top 12 similar images for the 21st image of the GHIM-10K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

(a) retrieval results of the MSD method [1]



(b) retrieval results of the proposed method

Figure 29: The top 12 similar images for the 6027th image of the GHIM-10K dataset retrieved by the (a) MSD method [1] and (b) proposed method.

MSD method. Table 5 lists the ARP and ARR comparison of different methods on the GHIM-10K dataset. The value of NR is 12. The highest ARP and ARR are highlighted in bold. Compared with all the methods listed here, our method performed the best.

Table 5: Comparison of different image retrieval methods on the GHIM-10K dataset. The value of NR is 12. The bold values indicate the best results.

Performance	MTH [44]	MSD [1]	CDH [2]	CH-LDP [50]	Liu2016 [63]	Liu2015 [64]	SSH [4]	CH-LDP-SIFT[51]	CH-LDP-SIFT+DP [51]	OUR
ARP	53.4	53.4	50.1	60.0	56.7	57.5	61.2	66.3	67.1	**68.5**
ARR	1.28	1.28	1.20	1.44	1.36	1.38	1.47	1.59	1.61	**1.65**

6.2.5. CIFAR-10 dataset

Table 6: The mAP@1000 of our method and some unsupervised hashing methods (64 hash bits) on the CIFAR-10 dataset. The results on the top, middle, and bottom sections are from shallow learning-based hashing methods, deep learning-based hashing methods, and our proposed method, respectively.

	Model	mAP@1000
Shallow	KMH [65]	14.46
	SphH [66]	13.98
	SpeH [67]	12.55
	PCAH [68]	12.10
	PCA-ITQ [69]	16.64
Deep	DH [70]	16.69
	BA [71]	28.8
	DeepBit [72]	27.73
	DBD-MQ [73]	31.85
	UTH [74]	32.41
	BDH [75]	25.9
	EDH [75]	30.8
	UDN2H [75]	27.2
	UDN2H_ae [75]	30.5
	UDN2H_org [75]	31.1
	USDH [76]	39.27
	ClusterGAN [77]	**45.12**
OUR		26.14

We test our method on the CIFAR-10 dataset [6] to evaluate the performance of our method on relative large datasets and to compare our method with some learning-based methods. The CIFAR-10 dataset is a labeled subset of the 80 tiny million tiny image collection. This dataset contains 60,000 color images in 10 categories. Each category consists of 6000 images. The size of each image is 32×32. Table 6 shows the mAP at top 1000 (i.e., mAP@1000) of different unsupervised hashing methods and our method on the CIFAR-10 dataset. The number of hash bits is 64 for all the results of the hashing methods listed here. The results on the top and middle sections are from the shallow and deep learning-based hashing methods, respectively. The result of our method is listed at the bottom section. Our method performed worse than most deep learning-based hashing methods listed in the table because it only employed low-level features to represent an image. Nevertheless, our method performed better than all the listed shallow learning-based hashing methods. Our method is easier to implement and is independent of the learning process that takes much processing time.

6.2.6. Comparison with other fusion-based methods

The proposed method is a fusion-based method. It represents an image by fusing the MMSD and uniform LBP features. To illustrate the effectiveness of the proposed method, we would like to compare it with some other fusion-based methods on the Corel-1K, Corel-5K, and Corel-10K datasets. Tables 7 and 8 list the ARP/ARR of the top 10 and 12 retrieved images, respectively. The first column shows the length of the feature vector defined in each method. The last three columns list the results on the three datasets. The proposed method had a relatively longer image descriptor but achieved the highest ARP/ARR compared with all the methods listed here. Reducing the length of the descriptor is one of our future tasks.

Given an image dataset, the proposed method computes a dissimilarity matrix $\boldsymbol{D_1}$ on the basis of the images' MMSDs and a dissimilarity matrix $\boldsymbol{D_2}$ on the basis of the images' uniform LBPs. The method then fuses the two

Table 7: Comparison (ARP/ARR, %) of different fusion-based methods on the Corel-1K, Corel-5K and Corel-10K datasets. The value of NR is 10.

Method	Feature length	Corel-1K	Corel-5K	Corel-10K
LTrP+LNDP [21]	1023	-/-	-/-	41.1/-
LBP+LNDP [21]	512	-/-	-/-	42.8/-
8D-GLCM+GSF [52]	40	60.9/-	42.3/-	32.7/-
GSF+HSVCM [52]	17	67.4/-	49.6/-	36.9/-
8D-GLCM+GSF+HSVCM [52]	49	68.8/-	54.2/-	40.3/-
FIF-IRS [52]	41	**83.3/-**	66.9/-	56.4/-
CH-LDP-SIFT [51]	220	-/-	65.7/-	56.0/-
CH-LDP-SIFT+DP [51]	220	-/-	65.9/-	57.0/-
Bag of Textual words + Bag of Visual words [57]	-	78.0/-	-/-	-/-
OUR	203	**83.3/8.3**	**70.2/7.0**	**61.0/6.1**

Table 8: Comparison (ARP/ARR, %) of different fusion-based methods on the Corel-1K, Corel-5K and Corel-10K datasets. The value of NR is 12.

Method	Feature length	Corel-1K	Corel-5K	Corel-10K
BDIP+BVLC (brightness) [49]	32	-/-	36.7/-	27.4/-
BDIP+BVLC (color) [49]	96	-/-	48.1/-	37.2/-
BDIP+BVLC (brightness) +CH [49]	96	-/-	57.5/-	47.0/-
BDIP+BVLC (color)+CH [49]	160	-/-	60.5/-	48.8/-
Orientation+Intensity [53]	16	57.5/6.9	37.1/4.5	28.0/3.4
Color+Orientation [53]	78	76.0/9.1	60.8/7.3	47.9/5.7
Color+Intensity [53]	82	77.2/9.2	61.8/7.4	50.1/6.0
Color+Orientation+Intensity [53]	88	78.5/9.4	63.1/7.6	50.3/6.0
IVD [54]	115	-/-	66.9/8.0	56.9/6.8
OUR	203	**82.2/9.9**	**67.9/8.2**	**58.5/7.0**

dissimilarity matrices to a similarity matrix $\boldsymbol{S}$. Updating $\boldsymbol{S}$ with a WAS, it obtains the final similarity matrix $\boldsymbol{S'}$. Except for fusing the dissimilarity matrix $\boldsymbol{D_1}$ with $\boldsymbol{D_2}$, $\boldsymbol{D_1}$ can be fused with a dissimilarity matrix computed on the basis of any other image descriptors. We explored the performance of fusing $\boldsymbol{D_1}$ with the dissimilarity matrix computed on the basis of the following seven image descriptors, namely, refined LBP [78], LDP [79], refined LDP [78], LTrP [80], refined LTrP [78], CDH [2], and MTH [44]. This undertaking is conducted to demonstrate the effectiveness of the proposed MMSD and WAS. For each descriptor, we use "MSD+Descriptor" to represent the method of fusing MSD and this descriptor, "MMSD+Descriptor" to represent the method of fusing MMSD and this descriptor, and "MMSD+Descriptor+WAS" to represent the method of applying WAS for updating the similarity matrix computed on the basis of MMSD+Descriptor. Table 9 shows the length of the descriptor used in these methods. We retrieve the top 12 similar images for the images within the Corel-1K, Corel-5K, and Corel-10K datasets. Tables 10, 11, and 12 show the results on these three datasets, respectively. For each descriptor shown in the left column, the second to the fifth columns show the ARP/ARR generated by only using the descriptor, MSD+Descriptor, MMSD+Descriptor and MMSD+Descriptor+WAS. The highest ARP/ARR is highlighted in bold. The results shown in these tables demonstrate that MSD+Descriptor performed better than the descriptor alone. Moreover, MMSD+Descriptor performed better than MSD+Descriptor, and MMSD+Descriptor+WAS performed the best. These results indicate that the proposed MMSD and WAS are effective for image retrieval.

6.3. Discussion

6.3.1. Determining the value of parameter k

After the dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ generated by comparing the images' MMSDs and uniform LBP histograms are combined, our method achieves a similarity matrix $\boldsymbol{S}$ of the images within the dataset. Our method then updates $\boldsymbol{S}$ by considering the relationships between the images and the top k

Table 9: Feature length of the image descriptor used in different methods.

Descriptor	Feature length			
	Descriptor	MSD+ Descriptor	MMSD+ Descriptor	MMSD+ Descriptor+ WAS
Refined LBP [78]	118	190	262	262
LDP [79]	236	308	380	380
Refined LDP [78]	472	544	616	616
LTrP [23]	767	839	911	911
Refined LTrP [78]	1534	1606	1678	1678
CDH [2]	108	180	252	252
MTH [44]	81	153	225	225

Table 10: ARP/ARR generated by some descriptors, MSD + each descriptor, MMSD + each descriptor, and MMSD + each descriptor + WAS on the Corel-1K dataset ($NR = 12$).

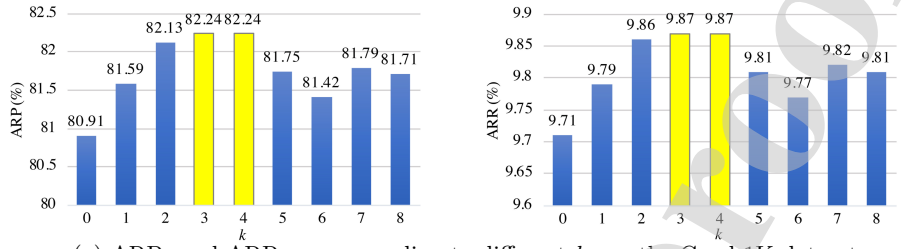
Descriptors	Descriptor	MSD + Descriptor	MMSD + Descriptor	MMSD + Descriptor + WAS
Refined LBP [78]	72.54/8.71	80.63/9.68	81.99/9.84	**83.58/10.03**
LDP [79]	67.5/8.1	79.71/9.57	81.08/9.73	**82.83/9.94**
Refined LDP [78]	71.30/8.56	80.88/9.71	81.70/9.80	**83.68/10.04**
LTrP [80]	63.78/7.65	76.08/9.13	77.84/9.34	**79.44/9.54**
Refined LTrP [78]	67.71/8.13	77.33/9.28	78.92/9.47	**80.45/9.65**
CDH [2]	65.75/7.89	79.64/9.56	80.42/9.65	**81.63/9.80**
MTH [44]	68.25/8.19	77.88/9.35	78.15/9.38	**80.04/9.61**

Table 11: ARP/ARR generated by some descriptors, MSD + each descriptor, MMSD + each descriptor, and MMSD + each descriptor + WAS on the Corel-5K dataset ($NR = 12$).

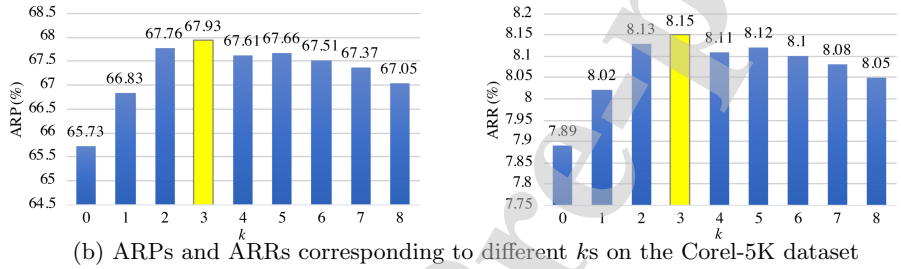
Descriptors	Descriptor	MSD + Descriptor	MMSD + Descriptor	MMSD + Descriptor + WAS
Refined LBP [78]	49.43/5.93	67.56/8.11	68.24/8.19	**70.71/8.49**
LDP [79]	45.86/5.5	66.84/8.02	68.1/8.17	**70.52/8.46**
Refined LDP [78]	49.52/5.94	67.08/8.05	68.04/8.17	**70.62/8.47**
LTrP [80]	39.99/4.8	64.08/7.69	65.23/7.83	**67.65/8.12**
Refined LTrP [78]	45.68/5.48	64.76/7.77	66.0/7.92	**68.34/8.2**
CDH [2]	57.23/6.87	66.19/7.94	66.83/8.02	**69.8/8.38**
MTH [44]	49.84/5.98	60.99/7.32	62.13/7.46	**64.37/7.72**

Table 12: ARP/ARR generated by some descriptors, MSD + each descriptor, MMSD + each descriptor, and MMSD + each descriptor + WAS on the Corel-10K dataset ($NR = 12$).

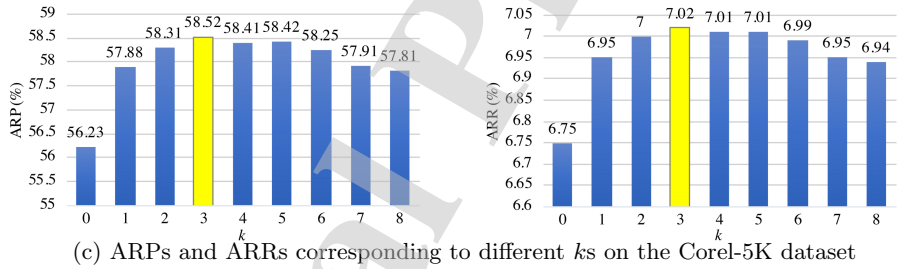
Refined LBP [78]	40.1/4.81	58.73/7.05	58.83/7.06	**61.84/7.42**
LDP [79]	37.38/4.49	57.91/6.95	58.05/6.97	**60.93/7.31**
Refined LDP [78]	40.33/4.84	58.53/7.02	58.62/7.03	**61.51/7.38**
LTrP [80]	32.2/3.86	55.38/6.65	56.04/6.72	**58.46/7.02**
Refined LTrP [78]	37.18/4.46	57.26/6.87	57.52/6.9	**59.53/7.14**
CDH [2]	45.24/5.43	52.8/6.34	53.17/6.38	**54.87/6.58**
MTH [44]	41.44/4.97	51.41/6.17	52.06/6.25	**53.98/6.48**



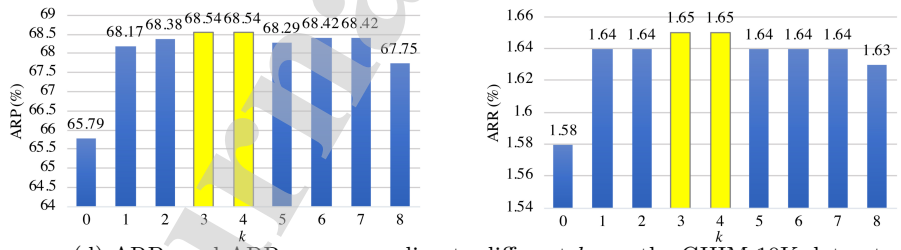
(a) ARPs and ARRs corresponding to different ks on the Corel-1K dataset



(b) ARPs and ARRs corresponding to different ks on the Corel-5K dataset



(c) ARPs and ARRs corresponding to different ks on the Corel-5K dataset



(d) ARPs and ARRs corresponding to different ks on the GHIM-10K dataset

Figure 30: ARPs and ARRs corresponding to different ks on the (a) Corel-1K, (b) Corel-5K, (c) Corel-10K, and (d) GHIM-10K datasets. The value of k varies from 0 to 8. The largest ARPs and ARRs are highlighted in yellow, and they are achieved when k is around 3.

similar images of these images. To determine the value of parameter k, we tested computing the ARPs and ARRs on the Corel-1K, Corel-5K, Corel-10K, and GHIM-10K datasets by setting k to different values within the rage $[0, 8]$. Figure 30 shows the ARPs and ARRs corresponding to different ks on the four datasets. The value of NR is 12. The highest ARPs and ARRs are achieved when k is around 3, and they are highlighted in yellow. Therefore, we set k to 3.

6.3.2. Time complexity

We consider the situation of retrieving similar images for a query image from a dataset containing N images. The size of each image is $U \times W$. To extract the feature of an image, our method first coverts the color space from RGB space to HSV space. The time complexity of this conversion is $O(U \times W)$. Our method then computes the orientation map θ for the image, and its time complexity is $O(U \times W)$. With the converted color image and orientation map, our method then computes the micro-structure maps M and M'. To construct M and M', our method considers the neighbors in four directions for each pixel, which has $O(4 \times U \times W)$ complexity. Two micro-structure images f and f' are then computed on the basis of M and M', and the time complexity of which is $O(U \times W)$. Comparing the values of f and f' at each pixel with those of f and f' at the pixel's eight neighbors, we achieve the MMSD of the image. Therefore, the total time complexity of extracting the MMSD of an image is $O(2 \times 8 \times U \times W)$. To capture the color difference information of the image, we further compute the uniform LBP histogram for the image by computing the color difference between each pixel and its eight neighbors. The time complexity of computing the uniform LBP histogram is $O(8 \times U \times W)$. We compute a dissimilarity vector $\boldsymbol{d_1}$ between the query image and the images within the dataset on the basis of the extracted MMSD features. The time complexity of comparing two images is $O(144)$, and that of computing $\boldsymbol{d_1}$ is $O(144 \times N)$ because the size of MMSD is 144. We compute a dissimilarity vector $\boldsymbol{d_2}$ between the query image and the images within the dataset on the

basis of the uniform LBP histograms. The time complexity of computing $\boldsymbol{d_2}$ is $O(59 \times N)$ because the dimension of the LBP histogram is 59. Our method then combines $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ and constructs a similarity vector $\boldsymbol{s}$, which has a time complexity of $O(N)$. Considering the relationship of the images and the images' top k similar images, our method then updates $\boldsymbol{s}$ to a new similarity vector $\boldsymbol{s'}$, and the time complexity of this process is $O(N \times k^2)$. Since we set k to 3, the time complexity of this process is $O(9 \times N)$. Based on the above discuss, the total time complexity of computing the similarity vector $\boldsymbol{s'}$ is $O(16{\times}N{\times}U{\times}W){+}O(8{\times}N{\times}U{\times}W){+}O(144{\times}N){+}O(59{\times}N){+}O(N){+}O(9{\times}N)$, i.e., $O(N \times U \times W)$. After computing the similarity vector $\boldsymbol{s'}$ between the query image and the images within the dataset, we use the merging sorting algorithm to sort the elements of $\boldsymbol{s'}$ in descending order. The time complexity of this sorting process is $O(N \log_2 N)$. The total time complexity of retrieving the similar images for a query image is $O(N \times U \times W) + O(N \log_2 N)$.

6.3.3. Other

Section 5.3 discusses the process of computing the similarity matrix $\boldsymbol{S}$. In this process, our method uses Eq. (15) to combine the dissimilarity matrices $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$. The parameter w in Eq. (15) represents the weight of $\boldsymbol{D_1}$. For all the experimental results shown in this study, our method simply sets the value of w to 0.5. Xu and Lu [81] proposed an adaptive weighted fusion method that could automatically determine the optimal weights without any manual setting in the field of score fusion. Inspired by this method [81], we would like to consider automatically determining the value of w as one of our future works.

7. Conclusion

In this work, we proposed a novel method for image retrieval. To describe an image, we modified the traditional MSD method to capture the direct relationship between the shape and texture features and the direct relationship between the color and texture features. We then applied the uniform LBP histogram to capture the local color difference of the pixels. To compare the images within

a dataset, we first computed the similarities of the images by comparing their descriptors. We further updated the similarity of each pair of images by also considering the similarities of their similar images within the dataset. Experimental results demonstrate the effectiveness of the proposed method on image retrieval. For future work, we would like to reduce the dimension of the image descriptor.

8. Acknowledgments

Funding: This work was supported by the Natural Science Foundation of Shandong Province [grant numbers ZR2019BF026, ZR2019MF013, ZR2017BF031]; the Project of Jinan Scientific Research Leader's Laboratory [grant number 2018GXRC023]; and the Doctoral Program of University of Jinan [grant number 160100313].

References

References

[1] G. Liu, Z. Li, L. Zhang, Y. Xu, Image retrieval based on micro-structure descriptor, Pattern Recognition 44 (9) (2011) 2123–2133. doi:10.1016/j.patcog.2011.02.003.

[2] G. Liu, J. Yang, Content-based image retrieval using color difference histogram, Pattern Recognition 46 (1) (2013) 188–198. doi:10.1016/j.patcog.2012.06.001.

[3] X. Wang, Z. Wang, A novel method for image retrieval based on structure elements' descriptor, Journal of Visual Communication & Image Representation 24 (1) (2013) 63–74. doi:10.1016/j.jvcir.2012.10.003.

[4] G. Liu, J. Yang, Z. Li, Content-based image retrieval using computational visual attention model, Pattern Recognition 48 (8) (2015) 2554–2566. doi:10.1016/j.patcog.2015.02.005.

[5] J. Z. Wang, L. Jia, G. Wiederholdy, SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, Pattern Analysis & Machine Intelligence IEEE Transactions on 23 (9) (2001) 947–963. doi:10.1109/34.955109.

[6] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto, Toronto (2009).

[7] A. Raza, H. Dawood, H. Dawood, S. Shabbir, R. Mehboob, A. Banjar, Correlated primary visual texton histogram features for content base image retrieval, IEEE Access 6 (99) (2018) 46595–46616. doi:10.1109/access.2018.2866091.

[8] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110. doi:10.1023/b:visi.0000029664.99615.94.

[9] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust features, in: European Conference on Computer Vision, 2006, pp. 404–417. doi:10.1007/11744023_32.

[10] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision & Pattern Recognition, 2005, pp. 886–893. doi:10.1109/CVPR.2005.177.

[11] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: IEEE International Conference on Computer Vision, 2003, pp. 1470–1477. doi:10.1109/ICCV.2003.1238663.

[12] D. Nistér, H. Stewénius, Scalable recognition with a vocabulary tree, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168. doi:10.1109/CVPR.2006.264.

[13] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 490–503. doi:10.1007/11744085_38.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587635.

[15] S. Josef, Z. Andrew, Efficient visual search of videos cast as text retrieval, IEEE Transactions on Pattern Analysis & Machine Intelligence 31 (4) (2009) 591–606. doi:10.1109/TPAMI.2008.111.

[16] V. Gemert, Jan C, C. J. Veenman, A. W. M. Smeulders, G. Jan-Mark, Visual word ambiguity, IEEE Transactions on Pattern Analysis & Machine Intelligence 32 (7) (2010) 1271–1283. doi:10.1109/TPAMI.2009.132.

[17] Y. Su, Improving image classification using semantic attributes, International Journal of Computer Vision 100 (1) (2012) 59–77. doi:10.1007/s11263-012-0529-4.

[18] T. Ojala, I. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1) (1996) 51–59. doi:10.1016/0031-3203(95)00067-4.

[19] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis & Machine Intelligence 24 (7) (2002) 971–987. doi:10.1109/TPAMI.2002.1017623.

[20] M. Heikkilä, M. Pietikäinen, C. Schmid, Description of interest regions with local binary patterns, Pattern Recognition 42 (3) (2009) 425–436. doi:10.1016/j.patcog.2008.08.014.

[21] M. Verma, B. Raman, Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval, Multimedia Tools & Applications 77 (2018) 11843–11866.

[22] S. Schwartz, Visual Perception: A Clinical Orientation, 4th edition, McGraw-Hill Medical, 2009.

[23] M. J. Swain, D. H. Ballard, Indexing via color histograms, in: International Conference on Computer Vision, 1990, pp. 11–32. doi:10.1109/ICCV.1990.139558.

[24] J. Han, K.-K. Ma, Fuzzy color histogram and its use in color image retrieval, IEEE Transaction On Image Processing 11 (8) (2002) 944–952. doi:10.1109/TIP.2002.801585.

[25] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of the 4th ACM Multimedia Conference, 1997, pp. 65–73. doi:10.1145/244130.244148.

[26] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 1997, pp. 762–768. doi:10.1109/CVPR.1997.609412.

[27] X. Wang, F. Li, Z. Chen, A fast fractal coding in application on image retrieval, Fractals 17 (4) (2009) 441–450. doi:10.1142/S0218348X09004557.

[28] B. Manjunath, J.-R. Ohm, V. V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology 11 (6) (2001) 703–715. doi:10.1109/76.927424.

[29] S. Zeng, R. Huang, H. Wang, Z. Kang, Image retrieval using spatiograms of colors quantized by Gaussian Mixture Models, Neurocomputing 171 (C) (2016) 673–684. doi:10.1016/j.neucom.2015.07.008.

[30] M. Subrahmanyam, Q. J. Wu, R. Maheshwari, R. Balasubramanian, Modified color motif co-occurrence matrix for image indexing and retrieval, Computers & Electrical Engineering 39 (3) (2013) 762–774. doi:10.1016/j.compeleceng.2012.11.023.

[31] N. Jhanwar, S. Chaudhuri, G. Seetharaman, B. Zavidovique, Content based image retrieval using motif cooccurrence matrix, Image and Vision Computing 22 (14) (2004) 1211–1220. doi:10.1016/j.imavis.2004.03.026.

[32] C. Wang, X. Wang, Y. Li, Z. Xia, C. Zhang, Quaternion polar harmonic Fourier moments for color images, Information Sciences 450 (2018) 141 – 156. doi:10.1016/j.ins.2018.03.040.

[33] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Transactions on Systems, Man and Cybernetics SMC-3 (6) (1973) 610–621. doi:10.1109/TSMC.1973.4309314.

[34] X. Wang, Z. Chen, J. Yun, An effective method for color image retrieval based on texture, Computer Standards & Interfaces 34 (1) (2012) 31–35. doi:10.1016/j.csi.2011.05.001.

[35] G. R. Cross, A. K. Jain, Markov random field texture models, IEEE Transactions on Pattern Analysis & Machine Intelligence 5 (1) (1983) 25–39. doi:10.1109/TPAMI.1983.4767341.

[36] B. S. Manjunath, W. Y. Ma, Texture features for browsing and retrieval of image data, IEEE Transactions on Pattern Analysis & Machine Intelligence 18 (8) (1996) 837–842. doi:10.1109/34.531803.

[37] Y. Pew-Thian, P. Raveendran, O. Seng-Huat, Image analysis using Hahn moments, IEEE Transactions on Pattern Analysis & Machine Intelligence 29 (11) (2007) 2057–2062. doi:10.1109/TPAMI.2007.70709.

[38] F. Mahmoudi, J. Shanbehzadeh, A. M. Eftekhari-Moghadam, H. Soltanian-Zadeh, Image retrieval based on shape similarity by edge orientation autocorrelogram, Pattern Recognition 36 (8) (2003) 1725–1736. doi:10.1016/s0031-3203(03)00010-4.

[39] C. Wang, X. Wang, Z. Xia, C. Zhang, Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm, Information Sciences 470 (2019) 109 – 120. doi:10.1016/j.ins.2018.08.028.

[40] I. Kunttu, L. Lepistö, J. Rauhamaa, A. Visa, Multiscale Fourier descriptor for shape-based image retrieval, in: Proceedings of the 17th International

Conference on Pattern Recognition, Vol. 2, 2004, pp. 765–768. doi:10.1109/ICPR.2004.1334371.

[41] L. Feng, J. Wu, S. Liu, H. Zhang, Global Correlation Descriptor: A novel image representation for image retrieval, Journal of Visual Communication & Image Representation 33 (2015) 104–114. doi:10.1016/j.jvcir.2015.09.002.

[42] S. R. Dubey, S. K. Singh, R. K. Singh, Rotation and scale invariant hybrid image descriptor and retrieval, Computers & Electrical Engineering 46 (C) (2015) 288–302. doi:10.1016/j.compeleceng.2015.04.011.

[43] G. Liu, J. Yang, Image retrieval based on the texton co-occurrence matrix, Pattern Recognition 41 (12) (2008) 3521–3527. doi:10.1016/j.patcog.2008.06.010.

[44] G. Liu, L. Zhang, Y. Hou, Z. Li, J. Yang, Image retrieval based on multi-texton histogram, Pattern Recognition 43 (7) (2010) 2380–2389. doi:10.1016/j.patcog.2010.02.012.

[45] X. Wang, Z. Wang, The method for image retrieval based on multi-factors correlation utilizing block truncation coding, Pattern Recognition 47 (10) (2014) 3293–3303. doi:10.1016/j.patcog.2014.04.020.

[46] S. Unar, X. Wang, C. Wang, M. Wang, New strategy for CBIR by combining low-level visual features with a colour descriptor, IET Image Processing 13 (7) (2019) 1191–1200. doi:10.1049/iet-ipr.2019.0098.

[47] M. Zhang, K. Zhang, Q. Feng, J. Wang, J. Kong, Y. Lu, A novel image retrieval method based on hybrid information descriptors, Journal of Visual Communication and Image Representation 25 (7) (2014) 1574–1587. doi:10.1016/j.jvcir.2014.06.016.

[48] A. Raza, T. Nawaz, H. Dawood, H. Dawood, Square texton histogram features for image retrieval, Multimedia Tools & Applications 78 (3) (2019) 2719–2746. doi:10.1007/s11042-018-5795-x.

[49] C. Singh, K. P. Kaur, A fast and efficient image retrieval system based on color and texture features, Journal of Visual Communication & Image Representation 41 (2016) 225–238. doi:10.1016/j.jvcir.2016.10.002.

[50] J. Zhou, X. Liu, T. Xu, J. Gan, W. Liu, A new fusion approach for content based image retrieval with color histogram and local directional pattern, International Journal of Machine Learning and Cybernetics 9 (2018) 677–689. doi:10.1007/s13042-016-0597-9.

[51] J. Zhou, X. Liu, W. Liu, J. Gan, Image retrieval based on effective feature extraction and diffusion process, Multimedia Tools & Applications 78 (2019) 6163–6190. doi:10.1007/s11042-018-6192-1.

[52] M. I. Thusnavis Bella, A. Vasuki, An efficient image retrieval framework using fused information feature, Computers & Electrical Engineering 75 (2019) 46–60. doi:10.1016/j.compeleceng.2019.01.022.

[53] H. Dawood, M. H. Alkinani, A. Raza, H. Dawood, R. Mehboob, S. Shabbir, Correlated microstructure descriptor for image retrieval, IEEE Access 7 (2019) 55206–55228. doi:10.1109/ACCESS.2019.2911954.

[54] Z. Wei, G. Liu, Image retrieval using the intensity variation descriptor, Mathematical Problems in Engineering (2020) 1–12.doi:10.1155/2020/6283987.

[55] S. Unar, X. Wang, C. Zhang, C. Wang, Detected text-based image retrieval approach for textual images, IET Image Processing 13 (3) (2019) 515–521. doi:10.1049/iet-ipr.2018.5277.

[56] S. Unar, X. Wang, C. Zhang, Visual and textual information fusion using Kernel method for content based image retrieval, Information Fusion 44 (2018) 176–187. doi:10.1016/j.inffus.2018.03.006.

[57] S. Unar, X. Wang, C. Wang, Y. Wang, A decisive content based image retrieval approach for feature fusion in visual and textual images, Knowledge-Based Systems 179 (2019) 8–20. doi:10.1016/j.knosys.2019.05.001.

[58] A. Irtaza, M. A. Jaffar, E. Aleisa, T. S. Choi, Embedding neural networks for semantic association in content based image retrieval, Multimedia Tools & Applications 72 (2) (2014) 1911–1931. doi:10.1007/s11042-013-1489-6.

[59] M. ElAlami, A new matching strategy for content based image retrieval system, Applied Soft Computing Journal 14 (2014) 407–418. doi:10.1016/j.asoc.2013.10.003.

[60] N. Ali, K. B. Bajwa, R. Sablatnig, S. A. Chatzichristofis, Z. Iqbal, M. Rashid, H. A. Habib, A novel image retrieval based on visual words integration of sift and surf 11 (6) (2016) e0157428. doi:10.1371/journal.pone.0157428.

[61] S. Uzma, M. Zahid, M. Toqeer, J. M. Arshad, R. Amjad, S. Tanzila, Scene analysis and search using local features and support vector machine for effective content-based image retrieval, Artificial Intelligence Review 52 (2019) 901–925. doi:10.1007/s10462-018-9636-0.

[62] V. P. Singh, R. Srivastava, Y. Pathak, S. Tiwari, K. Kaur, Content-based image retrieval based on supervised learning and statistical-based moments, Modern Physics Letters B 33 (3) (2019) 1950213. doi:10.1142/S0217984919502130.

[63] G. Liu, Content-based image retrieval based on cauchy density function histogram, in: International Conference on Natural Computation & Fuzzy Systems & Knowledge Discovery, 2016, pp. 506–510. doi:10.1109/FSKD.2016.7603225.

[64] G. Liu, Content-based image retrieval based on visual attention and the conditional probability, in: International Conference on Chemical, Material, and Food Engineering, 2015, pp. 838–842. doi:10.2991/cmfe-15.2015.199.

[65] K. He, F. Wen, J. Sun, K-Means Hashing: An affinity-preserving quantization method for learning binary compact codes, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 2938–2945. doi:10.1109/CVPR.2013.378.

[66] J. P. Heo, Y. Lee, J. He, S. F. Chang, S. E. Yoon, Spherical hashing, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 2957–2964. doi:10.1109/CVPR.2012.6248024.

[67] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.

[68] J. Wang, S. Kumar, S. F. Chang, Semi-supervised hashing for scalable image retrieval, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 3424–3431. doi:10.1109/CVPR.2010.5539994.

[69] Y. Gong, S. Lazebnik, Iterative quantization: A procrustean approach to learning binary codes, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 817–824. doi:10.1109/CVPR.2011.5995432.

[70] V. E. Liong, J. Lu, W. Gang, P. Moulin, Z. Jie, Deep hashing for compact binary codes learning, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015, pp. 2475–2483. doi:10.1109/CVPR.2015.7298862.

[71] M. A. Carreira-Perpinan, R. Raziperchikolaei, Hashing with binary autoencoders, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015, pp. 557–566. doi:10.1109/CVPR.2015.7298654.

[72] K. Lin, J. Lu, C. Chen, J. Zhou, Learning compact binary descriptors with unsupervised deep neural networks, in: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, 2016, pp. 1183–1192. doi:10.1109/CVPR.2016.133.

[73] Y. Duan, J. Lu, Z. Wang, J. Feng, J. Zhou, Learning deep binary descriptor with multi-quantization, in: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, 2017, pp. 1183–1192. doi:10.1109/CVPR.2017.516.

[74] S. Huang, Y. Xiong, Y. Zhang, J. Wang, Unsupervised triplet hashing for fast image retrieval, in: arXiv preprint, 2017. doi:10.1145/3126686.3126773.

[75] S. Chafik, M. A. El Yacoubi, I. Daoudi, H. El Ouardi, Unsupervised deep neuron-per-neuron hashing, Applied Intelligence 49 (2019) 2218–2232. doi:10.1007/s10489-018-1353-5.

[76] S. Jin, X. Sun, H. Yao, S. Zhou, Unsupervised semantic deep hashing, Neurocomputing 351 (2019) 19–25. doi:10.1016/j.neucom.2019.01.020.

[77] K. Ghasedi, X. Wang, C. Deng, H. Huang, Balanced self-paced learning for generative adversarial clustering network, in: Computer Vision and Pattern Recognition (CVPR), 2019 IEEE Conference on, 2019, pp. 4391–4400. doi:10.1109/CVPR.2019.00452.

[78] A. K. Tiwari, V. Kanhangad, R. B. Pachori, Histogram refinement for texture descriptor based image retrieval, Signal Processing: Image Communication 53 (2017) 73 – 85. doi:10.1016/j.image.2017.01.010.

[79] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Transaction on Image Processing 19 (6) (2010) 1635–1650. doi:10.1109/TIP.2010.2042645.

[80] S. Murala, R. P. Maheshwari, R. Balasubramanian, Local tetra patterns: a new feature descriptor for content-based image retrieval, IEEE Transactions on Image Processing 21 (5) (2012) 2874–2886. doi:10.1109/TIP.2012.2188809.

[81] Y. Xu, Y. Lu, Adaptive weighted fusion: A novel fusion approach for image classification, Neurocomputing 168 (2015) 566–574. doi:10.1016/j.neucom.2015.05.070.

Dongmei Niu: Conceptualization, Methodology, Validation, Writing - Original Draft, Funding acquisition

Xiuyang Zhao: Supervision, Funding acquisition

Xue Lin: Writing - Review & Editing, Funding acquisition

Caiming Zhang: Writing - Review & Editing

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: