Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8870467, 12 pages
https://doi.org/10.1155/2020/8870467

Research Article

Robust Image Hashing with Low-Rank Representation and Ring Partition

Zhenjun Tang, Zixuan Yu, Zhixin Li, Chunqiang Yu, and Xianquan Zhang

Guangxi Key Lab of Multi-Source Information Mining & Security, and Department of Computer Science, Guangxi Normal University, Guilin 541004, China

Correspondence should be addressed to Zhenjun Tang; tangzj230@163.com

Received 23 July 2020; Revised 15 August 2020; Accepted 19 August 2020; Published 1 September 2020

Academic Editor: Zhili Zhou

Image hashing has attracted much attention of the community of multimedia security in the past years. It has been successfully used in social event detection, image authentication, copy detection, image quality assessment, and so on. This paper presents a novel image hashing with low-rank representation (LRR) and ring partition. The proposed hashing finds the saliency map by the spectral residual model and exploits it to construct the visual representation of the preprocessed image. Next, the proposed hashing calculates the low-rank recovery of the visual representation by LRR and extracts the rotation-invariant hash from the low-rank recovery by ring partition. Hash similarity is finally determined by L_2 norm. Extensive experiments are done to validate effectiveness of the proposed hashing. The results demonstrate that the proposed hashing can reach a good balance between robustness and discrimination and is superior to some state-of-the-art hashing algorithms in terms of the area under the receiver operating characteristic curve.

1. Introduction

With the popularity of the platforms of the social network, such as Facebook and Twitter, more and more digital images are transmitted via the Internet and stored in the cyberspace. Therefore, efficient techniques are required for processing massive images and protecting content security. For example, when an important event happens, such as an opening ceremony of the Olympic Games, many people would like to forward the same image of the event in their social network. These forwarded images may undergo some digital operations, such as compression and enhancement. Consequently, there are many image copies of a hot event in the cyberspace. It is an important task to find a hot event of the social network by detecting image copies [1]. In recent years, a useful technique called image hashing [2, 3] attracts much attention of the community of multimedia security, which can extract a short code called hash based on the visual content of the input image regardless of its detailed bits. It can not only be applied to social event detection [1],but also can be used in many other applications [4–8], e.g., image retrieval, image authentication, image copy detection, and image quality assessment. In practice, the hash is used to represent the input image. As the hash is a short representation, the use of image hashing can achieve efficient data processing. In this paper, we study a novel hashing algorithm based on the low-rank representation model and ring partition.

The most important properties of the image hashing algorithm are robustness and discrimination [9]. The requirement of robustness is that the image hashing algorithm must map visually similar images to the same or similar hashes no matter whether their bit representations are the same or not. In other words, the hashing algorithm should be robust to normal digital operations, e.g., compression, filtering, and enhancement. This is because they alter the bit representations of digital images but keep their visual appearances unchanged. The requirement of discrimination is that the image hashing algorithm must map different images to completely different hashes. Since the number of different images is much bigger than that of similar images in practice, good discrimination means a low error rate of judging different images as similar images. This property is helpful to many applications, such as social event detection and image retrieval. Note that the two properties restrict each other. A high-performance algorithm should reach a good balance between them.

In the past years, many researchers have devoted themselves to developing image hashing algorithms. Some typical techniques are briefly introduced below. For example, Swaminathan et al. [10] combined the Fourier-Mellin transform and randomization method to develop secure image hashing. Monga and Evans [11] proposed to use feature points detected by the end-stopped wavelet to build the hash. These two hashing algorithms [10, 11] are resilient to small-angle rotation. Lv and Wang [12] combined SIFT features and Harris points to construct the hash. This scheme is robust to large-angle rotation and brightness adjustment, but its robustness against additive noise and blurring must be improved. Zhao et al. [13] derived the robust image hash for authentication from global features determined by Zernike moments and local features based on the position information of salient features. Tang et al. [14] calculated the image hash by jointly using the color vector angle (CVA) and discrete wavelet transform (DWT). Both the above hashing methods [13, 14] only resist small-angle rotation. Laradji et al. [15] extracted the hash of the color image by using the hypercomplex numbers and quaternion Fourier transform (QFT). This approach has good discrimination, but its robustness against rotation needs to be improved. Wang et al. [16] jointly exploited the Watson model and SIFT features to design the hashing algorithm for authentication. This method has better performance than the hashing method [13].

Recently, Yan et al. [17] introduced the quaternion techniques called quaternion Fourier transform and quaternion Fourier-Mellin moments to image hashing design. In another study, Yan et al. [18] exploited the adaptive local image features to design multiscale hashing. Both the hashing schemes [17, 18] reach good performance in tampering detection. Tang et al. [19] constructed the feature matrix via the DCT (discrete cosine transform) and learned hash code from the DCT-based matrix by local linear embedding. This hashing only resists image rotation within 5°. In [20], Tang et al. proposed to extract local statistical features from image rings and compressed statistical features by calculating vector distance. As the contents of image rings are rotation-invariant, rotation robustness of this hashing [20] is thus enhanced. Davarzani et al. [21] combined SVD (singular value decomposition) with CSLBP (Center-Symmetric Local Binary Patterns) to make the hashing scheme for authentication. The scheme is robust to additive noise and JPEG compression, but it is sensitive to rotation. Huang et al. [22] introduced the random walk to hash generation for improving security. In another work, Qin et al. [23] used SVD to conduct preprocessing and extracted hybrid features based on the circle-based feature and block-based feature to construct the hash. The hybrid feature-based hashing has good robustness against compression and filtering, but its computational cost should be reduced. Tang et al. [24] first proposed to construct a three-order tensor with image blocks and extracted the image hash from the three-order tensor by Tucker decomposition. This approach can resist small-angle rotation. Shen and Zhao [25] proposed to compute the image hash by using the color opponent component and quadtree structure features. This method shows good performance in image authentication, but it is fragile to image rotation.

From the above reviews, it can be found that many algorithms do not make a good balance between rotation robustness and discrimination. Aiming at this problem, we propose a new image hashing based on low-rank representation and ring partition. The main contributions of the proposed algorithm are as follows:

(1) We calculate the visual representation of the preprocessed image based on the saliency map extracted by the spectral residual model. Since the saliency map can indicate visual attention of human beings, the visual representation using the saliency map can effectively describe salient regions of the image. Consequently, hash generation with the visual representation can improve robustness of the proposed algorithm

(2) We propose to incorporate low-rank representation into ring partition. The low-rank representation can capture the global structure of data, which is helpful to make the discriminative hash. Since ring partition can produce a set of image rings invariant to rotation, hash code extraction based on image rings can reach good rotation robustness

Many experiments are done to validate effectiveness of the proposed algorithm. The results illustrate that the proposed algorithm can resist many digital operations, including image rotation with a large angle. Performance comparisons with some state-of-the-art algorithms are also done. The receiver operating characteristic (ROC) curve results show that the proposed algorithm has better classification performance than those of the compared algorithms in discrimination and robustness.

The structure of the rest of the paper is as follows. Section 2 describes the image hashing algorithm proposed in this paper. Sections 3 and 4 discuss experimental results and performance comparisons, respectively. Section 5 summarizes this paper.

2. Proposed Image Hashing

The proposed image hashing can be divided into four steps: preprocessing, visual representation calculation, low-rank representation, and ring partition. Figure 1 is the diagram of our algorithm. The preprocessing is to produce a normalized image, and the visual representation calculation is to generate an image representation which can indicate salient regions of the image. The use of low-rank representation is to extract principal image features for making the discriminative hash, and the use of ring partition can make the extracted feature code invariant to image rotation. These steps are described in detail in the following sections.

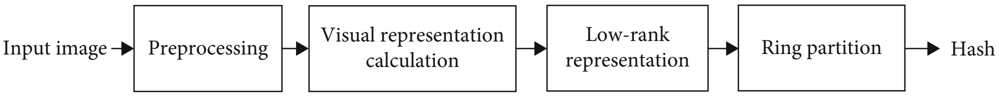


Figure 1: Diagram of the proposed hashing algorithm.

2.1. Preprocessing. This step includes three operations: bilinear interpolation, Gaussian low-pass filtering, and color space conversion. The bilinear interpolation is to resize the input image to a standard size $n \times n$. This operation can make our algorithm resilient to image scaling. The 3×3 Gaussian low-pass filtering is then applied to the resized image. Such operation can alleviate the influence of some digital operations on the resized image, such as image noise and JPEG compression. Finally, the filtered image in RGB color space is converted to HSV color space [26] and the brightness component in the HSV color space is used to denote the input image. The HSV color space is also called the hexagonal cone model and has shown good performances in many existing hashing algorithms [26]. Let H_1 be the hue of the pixel and S_1 and V_1 be the saturation and brightness of the pixel in the HSV color space, respectively. Thus, the calculation formulas for converting from RGB space to HSV space are as follows:

$$
\begin{aligned}
&H_1 = \begin{cases} \dfrac{(-B_1 + G_1) * \pi/3}{\operatorname{Max}(R_1, G_1, B_1) - \operatorname{Min}(R_1, G_1, B_1)} & \text{if } \mathrm{R}_1 = \operatorname{Max}(R_1, G_1, B_1), \\ \dfrac{(-R_1 + B_1) * \pi/3}{\operatorname{Max}(R_1, G_1, B_1) - \operatorname{Min}(R_1, G_1, B_1)} & \text{if } G_1 = \operatorname{Max}(R_1, G_1, B_1), \\ \dfrac{(-G_1 + R_1) * \pi/3}{\operatorname{Max}(R_1, G_1, B_1) - \operatorname{Min}(R_1, G_1, B_1)} & \text{if } B_1 = \operatorname{Max}(R_1, G_1, B_1), \\ \text{Undefined} & \text{if } R_1 = G_1 = B_1, \end{cases} \\
&S_1 = \begin{cases} \dfrac{\operatorname{Max}(R_1, G_1, B_1) - \operatorname{Min}(R_1, G_1, B_1)}{\operatorname{Max}(R_1, G_1, B_1)} & \text{if } \operatorname{Max}(R_1, G_1, B_1) \neq 0, \\ 0 & \text{if } \operatorname{Max}(R_1, G_1, B_1) = 0, \end{cases} \\
&V_1 = \operatorname{Max}(R_1, G_1, B_1),
\end{aligned}
\tag{1}
$$

where R_1 is the red component of the pixel, G_1 and B_1 are the green and blue components of the pixel, respectively, $\operatorname{Max}(R_1, G_1, B_1)$ and $\operatorname{Min}(R_1, G_1, B_1)$ represent the maximum value and the minimum value of R_1, G_1, and B_1, respectively.

Figure 2 shows a practical example of the preprocessing. Figure 2(a) is an input image, Figure 2(b) is the resized image, and Figure 2(c) represents the filtered image. Figures 2(d)–2(f) represent the hue, saturation, and brightness components of the filtered image in the HSV color space, respectively.

2.2. Visual Representation Calculation. In this work, a well-known model of saliency detection called the spectral residual model (SRM) [27] is exploited to find the saliency map of the image. Then, the visual image representation can be determined by combining the saliency map and the brightness component of the preprocessed image. Here, we select the SRM as the method of saliency detection. This is because the SRM is better than the conventional method such as Itti's method [28] in detection performance and computational speed [27].

Saliency map calculation of the classical spectral residual model is based on the spectral residual of the image $R(f)$, which is defined as follows:

$$R(f) = L(f) - h_n(f) * L(f), \tag{2}$$

where $h_n(f)$ is a matrix of size $n \times n$ and $L(f)$ is the log spectrum of the image. More specifically, $h_n(f)$ is defined as follows:

$$h_n(f) = \frac{1}{n^2} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{3}$$

In addition, $L(f)$ is determined by the below formula:

$$L(f) = \ln\,(A(f)), \tag{4}$$

where $A(f)$ is the amplitude spectrum of the image, which can be determined by the following formula:

$$A(f) = |F[I(x)]|, \tag{5}$$

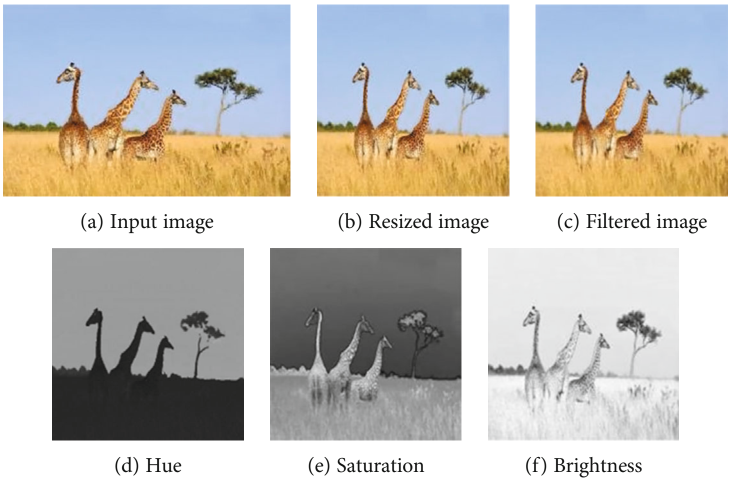
where $I(x)$ denotes a given image, $F[\bullet]$ is the Fourier transform, and $|\bullet|$ denotes the amplitude of the image. Finally, the saliency map $S(x)$ in the spatial domain can be constructed by using the inverse Fourier transform as follows:

$$S(x) = g(x) * F^{-1}[\exp\,(R(f) + P(f))]^2, \tag{6}$$

where $g(x)$ is a low-pass filter for smoothing the output saliency map of the inverse Fourier transform for better visual effects (a circular averaging filter radius 3 is used here), $F^{-1}[\bullet]$ is the inverse Fourier transform, and $P(f)$ is the phase spectrum of the image defined in the below equation:

$$P(f) = \varphi(F[I(x)]), \tag{7}$$

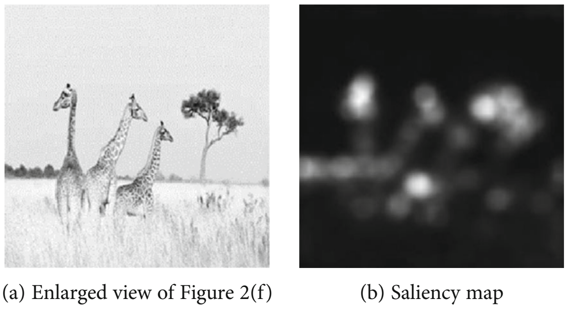
where $\varphi(\bullet)$ denotes the phase of the image. In [27], it is stated that image width (or height) with 64 pixels can reach a good estimation of the scale of normal visual conditions. Following this, we resize the $n \times n$ brightness component to 64×64 and convert the calculated saliency map to the original size $n \times n$ by bilinear interpolation. More details of SRM can be referred to [27].



(a) Input image (b) Resized image (c) Filtered image

(d) Hue (e) Saturation (f) Brightness

Figure 2: Practical example of preprocessing.



(a) Enlarged view of Figure 2(f) (b) Saliency map

Figure 3: An instance of saliency map detection via SRM.

Figure 3 shows an instance of saliency map detection via the spectral residual model. Figure 3(a) is the enlarged view of Figure 2(f), and Figure 3(b) is the saliency map of Figure 3(a) detected by the spectral residual model. From the results, it can be found that salient regions of the image are successfully extracted.

Suppose that $\mathbf{V}$ is the brightness component of the preprocessed image and $V(i,j)$ is the element of $\mathbf{V}$ in the ith row and the jth column. Also, assume that $\mathbf{S}$ denotes the saliency map of $\mathbf{V}$ and $S(i,j)$ is the element of $\mathbf{S}$ in the ith row and the jth column. Therefore, the visual representation $\mathbf{X}$ can be obtained by the following equation:

$$X(i,j) = V(i,j) \times S(i,j), \tag{8}$$

where $X(i,j)$ is the element of $\mathbf{X}$ in the ith row and the jth column.

2.3. Low-Rank Representation. Low-rank representation (LRR) is a useful technique for capturing the global structure of data [29]. The LRR is robust to noise and can extract the lowest-rank representation of all data [29, 30]. It has been widely used in many applications, such as subspace segmentation [31], image segmentation [32], and image classification [33]. Suppose that $\mathbf{X}$ is an observation matrix corrupted by noise $\mathbf{E}$. Thus, LRR calculation can be solved by the regularized rank minimization problem [29–31] as follows:

$$\min_{\mathbf{Z},\mathbf{E}} \quad \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{2,1}, \quad \text{s.t.} \quad \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \tag{9}$$

in which $\|\bullet\|_*$ is the nuclear norm of a matrix (sum of the singular values of the matrix), $\lambda > 0$ is a parameter for balancing effects of the two parts, $\|\bullet\|_{2,1}$ is the $l_{2,1}$ norm, and $\|\mathbf{E}\|_{2,1}$ is defined as follows:

$$\|\mathbf{E}\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n} \left([\mathbf{E}]_{ij}\right)^2}. \tag{10}$$

Let $\mathbf{Y}$ be the low-rank recovery of $\mathbf{X}$. Assume that the minimizer of (9) is $(\mathbf{Z}^*, \mathbf{E}^*)$. Thus, it can be obtained by $\mathbf{Y} = \mathbf{XZ}^*$ or $\mathbf{Y} = \mathbf{X} - \mathbf{E}^*$. In practice, problem (9) can be converted to an equivalent optimization problem as follows:

$$\min_{\mathbf{Z},\mathbf{E},\mathbf{J}} \quad \|\mathbf{J}\|_* + \lambda\|\mathbf{E}\|_{2,1}, \quad \text{s.t.} \quad \mathbf{X} = \mathbf{XZ} + \mathbf{E},\ \mathbf{Z} = \mathbf{J}. \tag{11}$$

This optimization problem can be solved by the below ALM (Augmented Lagrange Multiplier) problem:

$$\min_{\mathbf{Z,E,J,Y_1,Y_2}} \|\mathbf{J}\|_* + \lambda\|\mathbf{E}\|_{2,1} + \operatorname{tr}\left[\mathbf{Y}_1^{\mathrm{T}}(\mathbf{X}-\mathbf{XZ}-\mathbf{E})\right] + \operatorname{tr}\left[\mathbf{Y}_2^{\mathrm{T}}(\mathbf{Z}-\mathbf{J})\right] + \frac{\mu}{2}\left(\|\mathbf{X}-\mathbf{XZ}-\mathbf{E}\|_F^2 + \|\mathbf{Z}-\mathbf{J}\|_F^2\right), \tag{12}$$

in which $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are the Lagrange multipliers and $\mu > 0$ is the penalty parameter. Problem (12) can be solved by the inexact ALM method [30]. More details of LRR can be referred to [29–31].

In this work, the input of LRR is the visual representation $\mathbf{X}$ and the low-rank recovery $\mathbf{Y}$ is taken for hash code extraction. The reasons of our use of LRR for image hashing design are as follows. The influences of digital operations (e.g., compression, filtering, and noise) on image are viewed as noises added to the image. Since LRR is robust to noise, hash generation with low-rank recovery can improve robustness of our algorithm. In addition, LRR can efficiently capture the global structure of input data. Therefore, the use of LRR can ensure the discriminative capability of the proposed algorithm.

2.4. Ring Partition. To make the proposed algorithm resilient to image rotation, a well-known technique of image segmentation called ring partition (RP) [9, 34] is exploited here. RP takes the image center as the circle center and divides the inscribed circle of the image into a set of rings. Figure 4 presents an example of RP with 4 image rings. Clearly, the contents of image rings are unchanged after image rotation. Therefore, we can calculate the hash code resistant to rotation by using the mean values of these rings. Details of hash code extraction based on RP are explained as follows.

Suppose that the low-rank recovery $\mathbf{Y}$ is divided into m rings. Note that the size of $\mathbf{Y}$ is $n \times n$ and the area of each ring is kept the same. Obviously, the elements of image rings can be determined by using two adjacent radii except those of the innermost ring. Assume that r_i is the ith radius ($i = 1, 2, \cdots, m$) labeled from the small value to the big value. Therefore, r_1 is the radius of the innermost circle and r_m is the radius of the outmost circle. It is clear that $r_m = \lfloor n/2 \rfloor$, where $\lfloor \cdot \rfloor$ means downward rounding. To calculate the other radii, the average area of the image ring should be first determined by the below equation:

$$\mu_a = \lfloor C/m \rfloor, \tag{13}$$

where $C = \pi {r_m}^2$ is the area of the inscribed circle. Thus, the radius of the innermost circle can be calculated by the following equation:

$$r_1 = \sqrt{\frac{\mu_a}{\pi}}. \tag{14}$$

Next, other r_i ($i = 2, 3, \cdots, m-1$) can be determined by the below formula:

$$r_i = \sqrt{\frac{\mu_a + \pi r_{i-1}^2}{\pi}}. \tag{15}$$

After all radii are obtained, the elements of $\mathbf{Y}$ can be classified into image rings by using radii and the distances from these elements to the image center. Let $\mathbf{U}_i$ be the set of the elements of the ith ring ($i = 1, 2, \cdots, m$) and $p(x, y)$ be the element of $\mathbf{Y}$ in the xth row and yth column. Assume that the coordinates of the image center are (x_c, y_c). Therefore, $x_c = n/2 + 0.5$ and $y_c = n/2 + 0.5$ if n is an even number. Otherwise, $x_c = (n+1)/2$ and $y_c = (n+1)/2$. Thus, the distance from $p(x, y)$ to the image center (x_c, y_c) can be obtained by the below equation:

$$d_{x,y} = \sqrt{(x - x_c)^2 + (y - y_c)^2}. \tag{16}$$

Consequently, the set $\mathbf{U}_i$ can be determined by one of the following equations:

$$\begin{aligned} \mathbf{U}_1 &= \left\{p(x,y) \mid d_{x,y} \le r_1\right\}, \\ \mathbf{U}_i &= \left\{p(x,y) \mid r_{i-1} < d_{x,y} \le r_i\right\} \quad (i = 2, 3, \cdots, m). \end{aligned} \tag{17}$$

For each set, the mean of its elements is selected as compact feature. Let v_i be the mean of the elements in $\mathbf{U}_i$ ($i = 1, 2, \cdots, m$). Thus, it is quantized to an integer for reducing the storage by the below equation:

$$h(i) = [v_i \times 100 + 0.5] \quad (i = 1, 2, \cdots, m), \tag{18}$$

where $[\cdot]$ is the rounding operation. Finally, our hash is obtained by concatenating these integers as follows:

$$\mathbf{h} = [h(1), h(2), \cdots, h(m)]. \tag{19}$$

Therefore, the length of our hash is m integers.

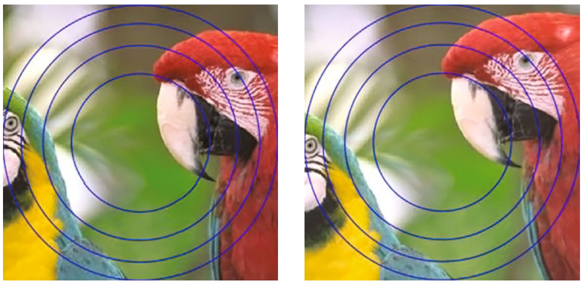
2.5. Hash Similarity Computation. As our hash is composed of some integers, the well-known distance metric called the L_2 norm is exploited to measure similarity between two hashes. Assume that $\mathbf{h}_1 = [h_1(1), h_1(2), \cdots, h_1(m)]$ and $\mathbf{h}_2 = [h_2(1), h_2(2), \cdots, h_2(m)]$ are the hash sequences of two images. Thus, the L_2 norm of the two hashes is defined as follows:

$$d_{\text{norm}} = \sqrt{\sum_{i=1}^{m} [h_1(i) - h_2(i)]^2}, \tag{20}$$

in which $h_1(i)$ is the ith element of $\mathbf{h}_1$ and $h_2(i)$ is the ith element of $\mathbf{h}_2$. Generally, a smaller L_2 norm means more similar hashes of the evaluated images. If the L_2 norm is bigger than a threshold T, the evaluated images corresponding to the input hashes are judged as different images. Otherwise, they are viewed as similar images.

3. Experimental Results

The parameter settings of the proposed algorithm are as follows. The λ of LRR is 0.9, the input image is resized to 512×512, and the ring number is 64, i.e., $n = 512$ and $m = 64$. In the following experiments, Sections 3.1 and 3.2



(a) An image (b) Rotated image with 10°

Figure 4: Example of ring partition with 4 rings.

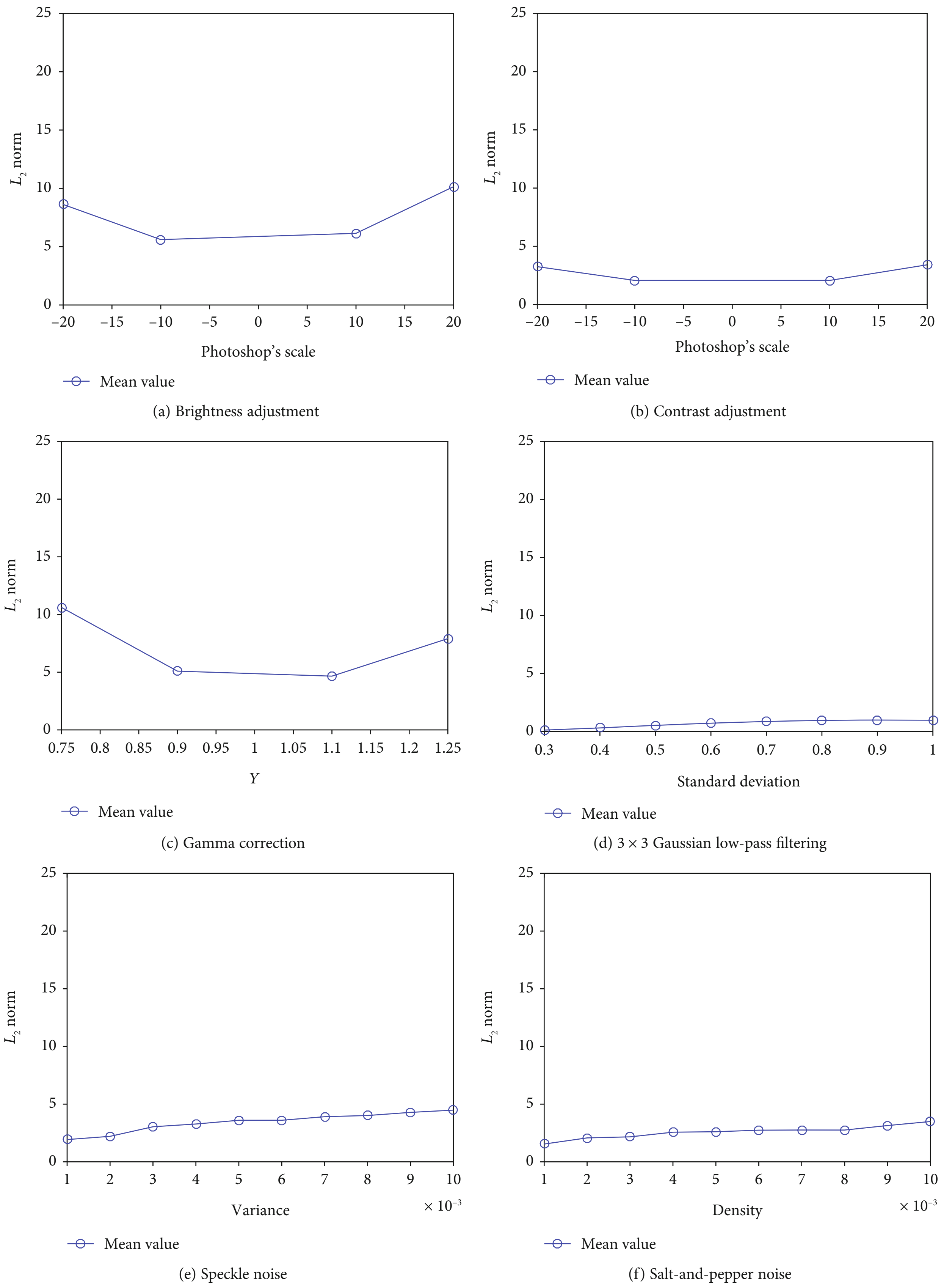
validate the properties of robustness and discrimination, respectively. Section 3.3 analyzes our hash storage in binary form. Section 3.4 discusses the influence of the ring number on our algorithm performance.

3.1. Robustness. An open database called the Kodak dataset [35] is exploited to construct a test database of similar images. The Kodak dataset is composed of 24 color images of different categories with the size of 768×512 or 512×768. To produce similar images of these color images, some commonly used operations are used to conduct robustness attacks. These operations are achieved by Photoshop, MATLAB, and StirMark. More specifically, Photoshop provides brightness and contrast adjustments (parameters are ±10 and ±20). MATLAB provides 3×3 Gaussian low-pass filtering (standard deviations range from 0.3 to 1.0 with a step 0.1), gamma correction (parameters are 0.75, 0.9, 1.1, and 1.25), salt-and-pepper noise, and speckle noise (both parameters range from 0.001 to 0.01 with a step 0.001). StirMark provides JPEG compression (quality factors range from 30 to 100 with a step 10), watermark embedding (strengths range from 10 to 100 with a step 10), image scaling (scaling ratios are 0.5, 0.75, 0.9, 1.1, 1.5, and 2.0), and image rotation (rotation angles are ±1, ±2, ±5, ±10, ±15, ±30, ±45, and ±90). Note that image rotation will increase image size and some padded pixels are added to the rotated images. In this experiment, only the 361×361 central parts of 24 original images and their rotated images are taken for evaluating rotation robustness. Therefore, the number of the used operations is 10, which totally contribute 80 manipulations. This implies that each original image has 80 similar images. So the total number of visual similar images is $24 \times 80 = 1920$, and the number of the used image is $1920 + 24 = 1944$.

Figure 5 demonstrates robustness experiments under different operations, where the x-axis is the parameter value of digital operation and the y-axis is the L_2 norm. Note that the curves in Figure 5 are the mean values of the L_2 norms between hashes of 24 color images and their similar images. From Figure 5, it can be seen that the mean L_2 norms are all smaller than 15, except two values of rotation operation. For image rotation, the maximum value is 17.29, which is a little bigger than those of other operations. It is found that, if the threshold is selected as $T = 15$, our algorithm can correctly detect 92.19% similar images. If there is no rotated image, our algorithm can recognize all similar images. A high correct detection rate illustrates good robustness of our algorithm.

3.2. Discrimination. A famous database named UCID [36] is selected to test discrimination of our algorithm. This database contains 1338 color images with the size of 512×384 or 384×512. Hashes of these 1338 color images are firstly calculated, and the L_2 norm between each pair of hashes is then computed. Therefore, the total number of valid distances is $C_{1338}^{2} = (1338 \times 1337)/2 = 894453$. Figure 6 illustrates the distribution of these distances, where the x-axis is the value of the L_2 norm and the y-axis is the frequency of the L_2 norm. It can be observed that the maximum L_2 norm is 163.25 and the minimum L_2 norm is 4.80. Moreover, the mean value of these L_2 norms is 42.72 and the standard deviation is 18.48. From Figure 6, it can be seen that most distances are bigger than the abovementioned threshold $T = 15$, indicating our good performance in discrimination. Actually, the performances of discrimination and robustness are closely related to the selected threshold. Different thresholds will lead to different performances. Table 1 demonstrates the performances of robustness and discrimination under different thresholds, where the robustness is measured by the correct detection rate and the discrimination is indicated by the false detection rate. Note that the correct detection rate is the ratio between the number of similar images correctly detected and the total number of similar images. The false detection rate is the ratio between the number of different images falsely judged as similar images and the total number of different images. From Table 1, we can select $T = 15$ as the recommended threshold since it can make the minimum total error rate.

3.3. Hash Storage. To analyze the required bits for storing our hash, the hashes of 1338 images in UCID are selected as the data source. Note that each hash generated by our algorithm is composed of 64 integers. Therefore, there are $1338 \times 64 = 85632$ integers in the data source. Figure 7 illustrates the distribution of these 85632 hash elements, where the x-axis is the element value and the y-axis is the frequency of the element value. It can be found that the



(a) Brightness adjustment

(b) Contrast adjustment

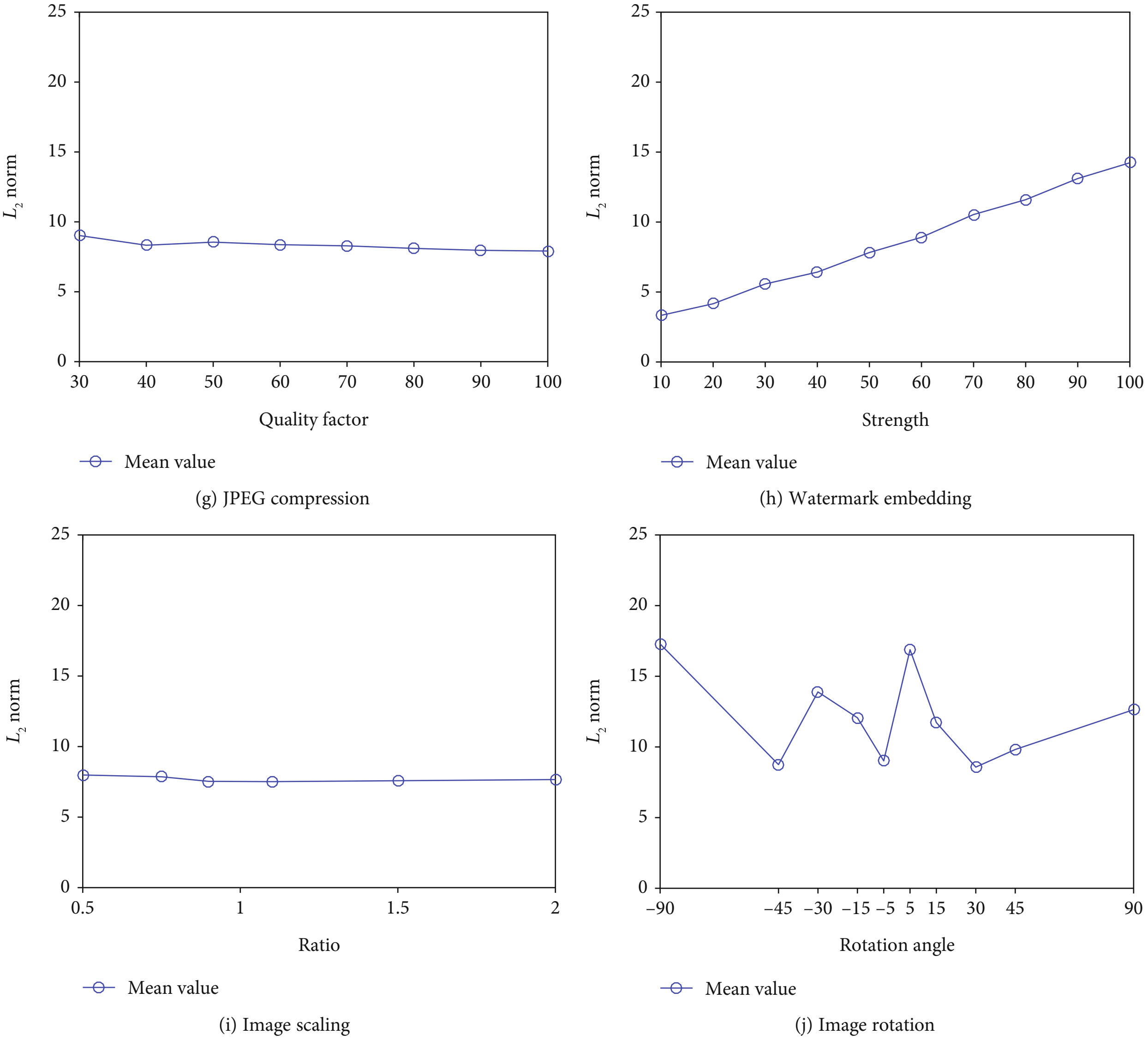
(c) Gamma correction

(d) 3 × 3 Gaussian low-pass filtering

(e) Speckle noise

(f) Salt-and-pepper noise

FIGURE 5: Continued.



(g) JPEG compression

(h) Watermark embedding

(i) Image scaling

(j) Image rotation

Figure 5: Robustness test.

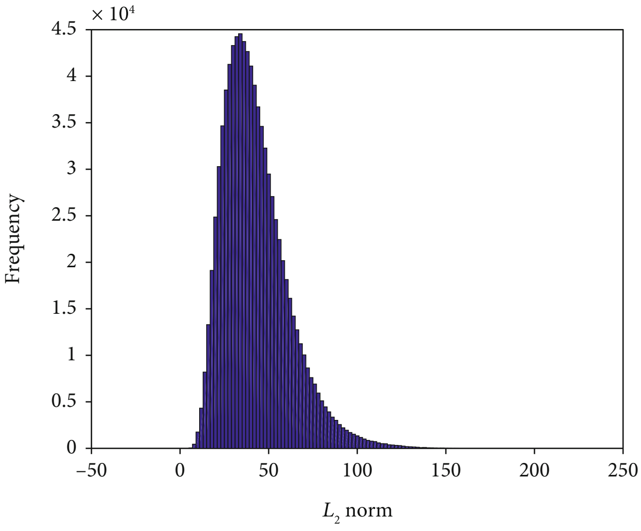


Figure 6: Distribution of L_2 norms based on 1338 images.

minimum value of these hash elements is 1 and the maximum value is 49. Since 6 bits can represent a decimal number in the range $[0, 2^6 - 1 = 63]$, the storage of a hash element only requires 6 bits. Therefore, our hash storage requires $64 \times 6 = 384$ bits in total.

3.4. Influence of the Ring Number on Hash Performance. To discuss the influence of the ring number on hash performance, different ring numbers are used to conduct robustness experiments and discriminative experiments. The used ring numbers include $m = 8$, $m = 16$, $m = 32$, $m = 64$, and $m = 80$. In the experiments, only the ring number is altered, and the other parameters are unchanged. The two image databases used in Sections 3.1 and 3.2 are also selected here. To make visual and quantitative comparisons, the receiver operating characteristic (ROC) graph [37] is taken. In the ROC graph, the x-axis represents FPR (false positive rate)

Table 1: Performances under different thresholds.

Threshold	Correct detection rate (d_1)	False detection rate (d_2)	Total error rate ($1 - d_1 + d_2$)
50	100%	70.89%	70.89%
45	99.74%	61.73%	61.99%
40	99.69%	50.97%	51.28%
35	99.58%	39.01%	39.43%
30	99.17%	26.64%	27.47%
25	98.39%	15.32%	16.93%
20	96.88%	6.58%	9.70%
15	92.19%	1.65%	9.46%
10	80.31%	0.13%	19.82%

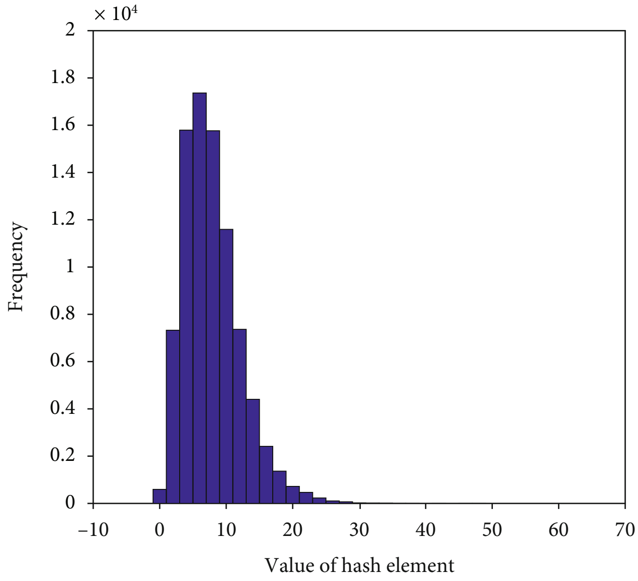


Figure 7: Distribution of 85632 hash elements.

and the y-axis represents TPR (true positive rate). Let P_{FPR} and P_{TPR} be the FPR and the TPR, respectively. Thus, they can be defined as follows:

$$P_{\mathrm{TPR}} = \frac{M_1}{M_2}, \quad P_{\mathrm{FPR}} = \frac{M_3}{M_4}, \tag{21}$$

in which M_1 is the number of similar images correctly judged as similar images, M_2 represents the total number of similar images, M_3 is the number of different images falsely identified as similar images, and M_4 denotes the total number of different images. Obviously, P_{FPR} and P_{TPR} can indicate discrimination and robustness, respectively. In the ROC graph, a curve is plotted by using a set of points with coordinates $(P_{\mathrm{FPR}}, P_{\mathrm{TPR}})$. High performances of discrimination and robustness mean a low P_{FPR} and a big P_{TPR}. Therefore, it can intuitively conclude that the curve near the top-left corner has better classification performance than the curve far away from it. To make quantitative analysis, the AUC (area under the ROC curve) is taken. The range of AUC is [0, 1]. In general, a bigger AUC means better classification of the evaluated algorithm.

Figure 8 shows ROC curves of different ring numbers, where the curves in the upper-left area are zoomed in for easy comparison. Obviously, the five ROC curves are all close to the upper-left corner, indicating good classification of the proposed algorithm. In addition, the curves of ring numbers $m = 64$ and $m = 80$ are overlapping and both of them are above the curves of the other ring numbers. This means that the ring numbers $m = 64$ and $m = 80$ are slightly better than other ring numbers in terms of classification performance between robustness and discrimination. To make quantitative analysis, the AUC of each ring number is also calculated. It is found that the AUCs of ring numbers $m = 8$, $m = 16$, $m = 32$, $m = 64$, and $m = 80$ are 0.98605, 0.98863, 0.99012, 0.99069, and 0.99087, respectively. Clearly, the AUCs of $m = 64$ and $m = 80$ are slightly different. They are all bigger than those of other ring numbers. This also verifies that $m = 64$ and $m = 80$ are better than other ring numbers.

To view computational time of different ring numbers, the proposed algorithm is coded by using MATLAB language and the used computer is a workstation with 2.10 GHz Intel Xeon Silver 4110 CPU and 64 GB RAM. The total time of extracting 1338 image hashes is computed to find the average time of generating a hash. It is found that the average time of $m = 8$, $m = 16$, $m = 32$, $m = 64$, and $m = 80$ is 25.407, 27.133, 26.957, 29.361, and 33.611 seconds, respectively. Moreover, hash lengths of different ring numbers are also compared. Note that our hash length is equal to the ring number when the length is measured by integers. Therefore, the hash lengths of $m = 8$, $m = 16$, $m = 32$, $m = 64$, and $m = 80$ are 8, 16, 32, 64, and 80 integers, respectively. Table 2 summarizes performances of different ring numbers. From the viewpoint of the whole performance, it can be observed that the ring number $m = 64$ reaches a good balance among the three performance indices.

4. Performance Comparisons

To demonstrate advantages of our hashing algorithm, we compare it with some popular hashing algorithms. The selected hashing algorithms are CVA-DWT hashing [14], SVD-CSLBP hashing [21], random walk-based hashing [22], and hybrid feature-based hashing [23]. The datasets used in the above robustness experiment and discrimination test are also taken here, i.e., 1920 pairs of similar images and 1338 different images. To make fair comparison, the parameters of the selected algorithms are set to the same values as their original papers. Their hash similarity metrics are kept unchanged, i.e., the L_2 norm for CVA-DWT hashing and hybrid feature-based hashing, the correlation coefficient for SVD-CSLBP hashing, and the normalized Hamming distance for random walk-based hashing. The result of our hashing with $m = 64$ is taken for comparison.

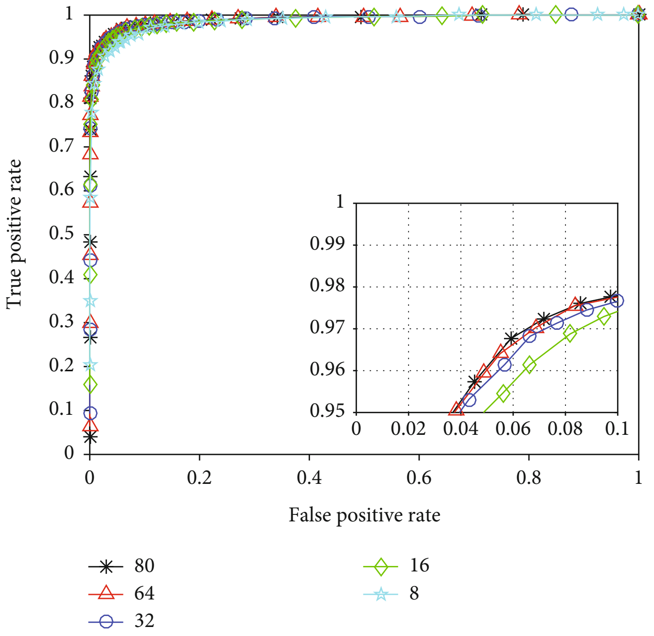


Figure 8: ROC curves of different ring numbers.

Table 2: Performances of different ring numbers.

m	AUC	Hash length (integer)	Computational time (s)
8	0.98605	8	25.407
16	0.98863	16	27.133
32	0.99012	32	26.957
64	0.99069	64	29.361
80	0.99087	80	33.611

Figure 9 illustrates the ROC curves of the evaluated hashing algorithms. To see more details, the ROC curves in the upper-left area are zoomed in and placed in the bottom-right part of Figure 9. It is observed that the curves of our hashing and hybrid feature-based hashing intersect with each other. Moreover, both of them are above the curves of other evaluated hashing algorithms. To conduct quantitative analysis, the AUCs of the evaluated hashing algorithms are also calculated. It is found that the AUCs of CVA-DWT hashing, SVD-CSLBP hashing, random walk-based hashing, hybrid feature-based hashing, and our hashing are 0.97563, 0.74522, 0.95758, 0.97545, and 0.99069, respectively. From the results, it can be seen that our hashing is better than the compared algorithms in classification performance.

Average time of calculating a hash is also compared. The abovementioned computer is used again, and the compared algorithms are also implemented with MATLAB. It is found that the average time of CVA-DWT hashing, SVD-CSLBP hashing, random walk-based hashing, hybrid feature-based hashing, and our hashing are 0.05, 0.19, 0.02, 6.12, and 29.36 seconds, respectively. The speed of our hashing is slower than those of the compared algorithms. This is because the computational cost of the LRR method is relatively high. Moreover, the hash lengths of all algorithms are also compared. The hash lengths of CVA-DWT hashing and random walk-based hashing are 960 and 144 bits, respectively. The hash lengths of SVD-CSLBP hashing and hybrid feature-based hashing are 64 and 104 floating-point numbers. As a floating-point number requires 32 bits for storage according to the IEEE standard [38], the hash lengths of SVD-CSLBP hashing and hybrid feature-based hashing are 2048 and 3328 bits, respectively. The length of our hashing is 384 bits. It is longer than that of random walk-based hashing, but it is much shorter than those of other compared hashing algorithms. Table 3 summarizes performance indices of the evaluated hashing algorithms, where the text in italic is the best result of the corresponding column. Our hashing outperforms the compared hashing algorithms in classification performance in terms of AUC, but it runs slower than the compared algorithms. As to hash length, it is better than all compared algorithms, except random walk-based hashing.

5. Conclusions

In this paper, we have proposed a novel image hashing with LRR and RP. An important contribution is the calculation of the visual representation based on the saliency map determined by SRM. Hash generation based on the visual representation can improve robustness performance. Another significant contribution is the combination of LRR and RP, which can make the discriminative hash invariant to rotation. Many experiments with two well-known databases have been carried out. The results have shown that the proposed hashing is robust and discriminative. ROC curve comparisons have illustrated that the proposed hashing outperforms the compared hashing algorithms in classification performance. In addition, hash length comparisons have shown

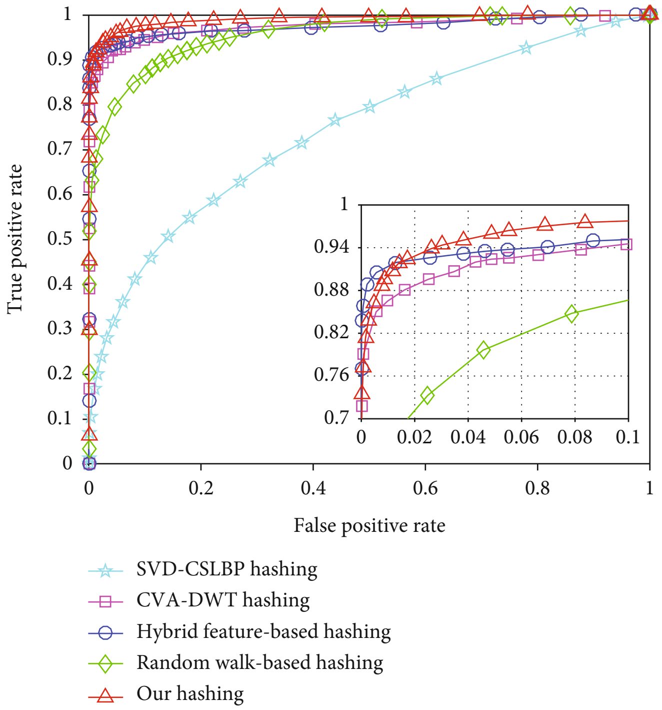


Figure 9: ROC curves of the evaluated hashing algorithms.

Table 3: Performance comparisons among different algorithms (text in italic is the best result).

Algorithm	AUC	Average time (s)	Hash length (bit)
CVA-DWT hashing	0.97563	0.05	960
SVD-CSLBP hashing	0.74522	0.19	2048
Random walk-based hashing	0.95758	*0.02*	*144*
Hybrid feature-based hashing	0.97545	6.12	3328
Our hashing	*0.99069*	29.36	384

that the proposed hashing is better than all compared algorithms, except random walk-based hashing. As to running speed, the proposed hashing runs slower than the compared algorithms due to the high computational cost of the LRR method. In the future, we plan to design fast hashing algorithms, deep learning-based hashing algorithms, and hashing algorithms for image authentication.

Data Availability

The image datasets used to support the findings of this study can be downloaded from the public websites whose hyperlinks are provided in this paper.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding publication of this paper.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (61962008, 61762017, and 61966004), the Guangxi “Bagui Scholar” Team for Innovation and Research, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Natural Science Foundation (2017GXNSFAA198222), the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, and the Innovation Project of Guangxi Graduate Education (YCSW2020109).

References

[1] P. J. McParlane, A. J. McMinn, and J. M. Jose, ““Picture the scene...”;:visually summarising social media events,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, pp. 1459–1468, Shanghai, China, 2014.

[2] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, “Robust image hashing,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, pp. 664–666, Vancouver, BC, Canada, 2000.

[3] Z. Tang, Z. Huang, X. Q. Zhang, and H. Lao, “Robust image hashing with multidimensional scaling,” *Signal Processing*, vol. 137, pp. 240–250, 2017.

[4] D. Wu, Z. Lin, B. Li, M. Ye, and W. Wang, “Deep supervised hashing for multilabel and large-scale image retrieval,” in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 150–158, Bucharest, Romania, 2017.

[5] Z. Tang, Z. Huang, H. Yao, X. Q. Zhang, L. Chen, and C. Yu, “Perceptual image hashing with weighted DWT features for

reduced-reference image quality assessment," *The Computer Journal*, vol. 61, no. 11, pp. 1695–1709, 2018.

[6] J. Ouyang, Y. Liu, and H. Shu, "Robust hashing for image authentication using SIFT feature and quaternion Zernike moments," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2609–2626, 2017.

[7] Z. Zhou, Q. M. J. Wu, Y. Yang, and X. Sun, "Region-level visual consistency verification for large-scale partial-duplicate image search," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 2, article 54, pp. 1–25, 2020.

[8] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[9] Z. Tang, X. Q. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.

[10] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, 2006.

[11] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, 2006.

[12] X. Lv and Z. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1081–1093, 2012.

[13] Y. Zhao, S. Wang, X. P. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55–63, 2013.

[14] Z. Tang, Y. Dai, X. Q. Zhang, L. Huang, and F. Yang, "Robust image hashing via colour vector angles and discrete wavelet transform," *IET Image Processing*, vol. 8, no. 3, pp. 142–149, 2014.

[15] I. Laradji, L. Ghouti, and E. Khiari, "Perceptual hashing of color images using hypercomplex representations," in *2013 IEEE International Conference on Image Processing*, pp. 4402–4406, Melbourne, VIC, Australia, 2013.

[16] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1336–1349, 2015.

[17] C. Yan, C. Pun, and X. Yuan, "Quaternion-based image hashing for adaptive tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2664–2677, 2016.

[18] C. Yan, C. Pun, and X. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Processing*, vol. 121, pp. 1–16, 2016.

[19] Z. Tang, H. Lao, X. Q. Zhang, and K. Liu, "Robust image hashing via DCT and LLE," *Computers & Security*, vol. 62, pp. 133–148, 2016.

[20] Z. Tang, X. Q. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.

[21] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools and Application*, vol. 75, no. 8, pp. 4639–4667, 2016.

[22] X. Huang, X. Liu, G. Wang, and M. Su, "A robust image hashing with enhanced randomness by using random walk on zigzag blocking," in *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 14–18, Tianjin, China, 2016.

[23] C. Qin, M. Sun, and C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Processing*, vol. 142, pp. 194–205, 2018.

[24] Z. Tang, L. Chen, X. Q. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.

[25] Q. Shen and Y. Zhao, "Perceptual hashing for color image based on color opponent component and quadtree structure," *Signal Processing*, vol. 166, article 107244, 2020.

[26] Z. Tang, X. Li, J. Song, M. Wei, and X. Q. Zhang, "Colour space selection in image hashing: an experimental study," *IETE Technical Review*, vol. 34, no. 4, pp. 440–447, 2016.

[27] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, 2007.

[28] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[29] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 663–670, Haifa, Israel, 2010.

[30] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[31] J. Chen and J. Yang, "Robust subspace segmentation via low-rank representation," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1432–1445, 2014.

[32] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *2011 International Conference on Computer Vision*, pp. 2439–2446, Barcelona, Spain, 2011.

[33] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 911–923, 2019.

[34] Z. Tang, X. Q. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Processing*, vol. 93, no. 7, pp. 2061–2069, 2013.

[35] "Kodak lossless true color image suite," June 2020, http://r0k.us/graphics/kodak/.

[36] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," in *Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307, pp. 472–480, San Jose, California, USA, 2004.

[37] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[38] IEEE Std754–2008, *IEEE Standard for Floating-Point Arithmetic*, pp. 1–70, IEEE, 2008.