Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 6103–6120.
DOI: 10.3934/mbe.2019305
Received: 06 March 2019
Accepted: 24 June 2019
Published: 03 July 2019

Research article

Robust image hashing via visual attention model and ring partition

Zhenjun Tang*, Yongzheng Yu, Hanyun Zhang, Mengzhu Yu, 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:** Email: tangzj230@163.com; Tel: +86-773-581-1621.

Abstract: Robustness is an important property of image hashing. Most of the existing hashing algorithms do not reach good robustness against large-angle rotation. Aiming at this problem, we jointly exploit visual attention model and ring partition to design a novel image hashing, which can make good rotation robustness. In the proposed image hashing, a visual attention model called PFT (Phase spectrum of Fourier Transform) model is used to detect saliency map of preprocessed image. The LL sub-band of saliency map is then divided into concentric circles invariant to rotation by ring partition, and the means and variances of DWT coefficients on concentric circles are taken as image features. Next, these features are encrypted by a chaotic map and the Euclidean distances between normalized encrypted features are finally exploited to generate hash. Similarity between hashes is measured by L_1 norm. Many experimental tests show that our image hashing is robust to digital operations including rotation and reaches good discrimination. Comparisons demonstrate that classification performance of our image hashing outperforms those of some well-known hashing algorithms in terms of receiver operating characteristics curves. Simulation of image copy detection is carried out on an open image database called UCID and the result validates effectiveness of our hashing.

Keywords: image hashing; visual attention model; ring partition; saliency map; image copy detection

1. Introduction

Massive images are generated in the big data era and shared by users via the Internet. In the massive images, some images are the copies of other images, which undergo some digital operations, such as compression and image enhancement. In general, image copies are visually similar with their original images, but digital data of their files are different. It is a challenge task to quickly identify image copies of a given image from a large-scale image database. In recent years, a new emerging multimedia technique called image hashing [1] can efficiently detect image copy from large-scale image database. Image hashing maps an input image to a short representation called image hash according to its visual content instead of the digital data of its file. Therefore, it can produce the same or similar hashes for visual similar images. This is the robustness property of image hashing [2]. For different input images, image hashing can convert them to different image hashes. This is the discrimination property of image hashing [3]. Besides the application of image copy detection, image hashing can be also applied to image authentication and image forensics. For these applications, image hashing should meet security property [4], which requires that hash generation must be controlled by secret key and different input keys should lead to different hashes of the same input image. In this paper, we study a novel image hashing based on visual attention model and ring partition.

In the past years, image hashing has attracted much attention from multimedia community. Many researchers have proposed various useful techniques to make a rapid development of image hashing. For example, Lefebvre et al. [5] proposed to use Radon transform (RT) to construct hash for resisting geometric transforms. Swaminathan et al. [6] exploited two-dimensional (2-D) DFT (Discrete Fourier Transform) in polar coordinates and random encryption with pseudorandom numbers to make secure hash. Monga and Evans [7] selected end-stopped wavelet transform to find visual feature points for constructing hash resilient to moderate rotation. Ou and Rhee [8] jointly used RT, one dimensional DCT (Discrete Cosine Transform) and data permutation to form secure hash. Tang et al. [9] used correlation coefficient to extract structural features for applying image hash to the application of tampering detection. Qin et al. [10] exploited non-uniform sampling in DFT domain to construct hash for resisting image rotation. Tang et al. [11] selected image entropies to measure local image textures and used them to make rotation-resistant hash. In another work, Tang et al. [12] utilized color vector angle (CAV) and 2-D discrete wavelet transform (DWT) to extract discriminative hash. To improve discrimination, Ghouti [13] applied quaternion singular value decomposition to hash extraction. Recently, Yan et al. [14] exploited quaternion Fourier-Mellin transform to develop hashing for tampering localization. Qin et al. [15] used selective sampling and structure features to construct perceptual hash for image authentication. Tang et al. [16] exploited ring partition to extract perceptual features and compressed them to make compact hash by invariant vector distance. Karsh et al. [17] combined global and local features to form image hash. In another work, Karsh et al. [18] jointly used DWT, SVD (Singular Value Decomposition) and spectral residual model to design robust hash. Davarzani et al. [19] computed image hash by using SVD and CSLBP (center-symmetric local binary patterns). Huang et al. [20] applied random walk to hash generation for improving security. Karsh et al. [21] extracted global and local features via ring partition and Markov abortion probabilities to construct hashing for authentication. Qin et al. [22] used circle-based strategy and the block-based strategy to extract hybrid features for hash construction. Tang et al. [23] selected weighted DWT features to make perceptual hash for image quality assessment.

Most of the existing image hashing algorithms do not reach good robustness against large-angle rotation without decreasing discrimination. To overcome this problem, we jointly exploit visual attention model and ring partition to design a novel image hashing. Compared with the existing techniques, our image hashing has two contributions as follows.

(1) A visual attention model called PFT (Phase spectrum of Fourier Transform) model is exploited to detect saliency map of input image. As salient areas can indicate the regions of attention focus in human eye, hash construction based on salient areas can improve perceptual robustness of

our image hashing.

(2) Ring partition is applied to saliency map in DWT domain. Since ring partition can divide LL sub-band into some concentric circles invariant to image rotation, statistical features extracted from concentric circles can resist large-angle rotation.

Experiments are done to validate performances of the proposed algorithm. The results illustrate that the proposed algorithm can resist large-angle rotation and reach good discrimination. Comparisons show that the proposed algorithm outperforms some state-of-the-art algorithms. The rest of this paper is organized as follows. Section 2 explains details of the proposed algorithm. Section 3 presents experimental results and Section 4 discusses our application to image copy detection. Section 5 concludes this paper.

2. Proposed image hashing

Our proposed algorithm consists of four steps, i.e., preprocessing, saliency map detection, ring partition, data encryption and compression. In the preprocessing, bi-cubic interpolation is first exploited to convert input image to a fixed size *S*×*S* and then the luminance component Y in YCbCr color space is taken for representing color image itself. We select the luminance component for calculation. This is because it contains most information of color image and the input of the model of saliency map detection used in this paper is gray-scale image. Conversion from RGB color space to YCbCr color space can be done by the below equation.

$$\begin{bmatrix} Y \\ C_\mathrm{b} \\ C_\mathrm{r} \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1}$$

where *Y*, C_b and C_r represent luminance component, blue-difference chroma component and red-difference chroma component, *R*, *G* and *B* are the red, green and blue components of a pixel, respectively. Details of other steps are explained in the following sections.

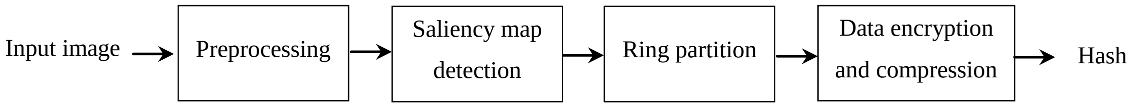


Figure 1. Schematic diagram of the proposed image hashing.

2.1. Saliency map detection

Salient areas can indicate the regions of attention focus in human eye. To improve perceptual robustness of the proposed image hashing, we exploit visual attention model to detect saliency map and use it to construct image hash. Researchers have developed some useful visual attention models, such as NVT (Neuromorphic Vision Toolkit) model [24], STB (SaliencyToolBox) model [25], SR (Spectral Residual) model [26], and PFT (Phase spectrum of Fourier Transform) model [27]. In this paper, we select the PFT model to conduct saliency map detection since it runs fast and is more effective than NVT model, STB model and SR model. Let *I*(*x*, *y*) be an input image. Suppose that *F* and F^{-1} represent Fourier transform and inverse Fourier transform, respectively. Thus, the classical PFT model can detect saliency map by the following equation [27].

$$M(x,y) = g(x,y) * \left\| F^{-1}[e^{i \cdot p(x,y)}] \right\|^2 \tag{2}$$

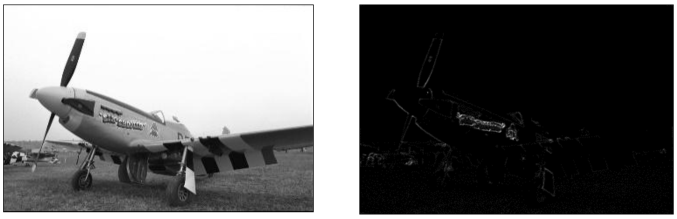
where $g(x, y)$ is a two-dimensional Gaussian filter with σ=8 (the reported value in [27]) and $p(x, y)$ is determined by the below equation.

$$p(x,y) = P(f(x,y)) \tag{3}$$

in which $P(\cdot)$ denotes the phase spectrum of the input image, and $f(x, y)$ is defined as follows.

$$f(x,y) = F(I(x,y)) \tag{4}$$

Figure 2 presents an example of detecting saliency map via PFT model. Figure 2 (a) is a gray image and Figure 2 (b) is the saliency map detected by PFT model. From the result, it can be seen that the detected result is consistent with the attention focus of human visual system.



(a) a gray image (b) Saliency map

Figure 2. An example of saliency map detection via PFT model.

2.2. Ring partition

A single level 2D DWT is applied to the detected saliency map and the LL sub-band is taken for representation. This operation can reduce the influence of image noise on the saliency map and reach initial data compression (the number of the coefficients in LL sub-band is only 1/4 of the total DWT coefficients). To make our hash resistant to large-angle rotation, we exploit ring partition [28] to divide the LL sub-band into several concentric circles. As concentric circles are invariant to image rotation [29], hash construction with DWT coefficients on the concentric circles can reach good robustness against rotation. Extraction of DWT coefficients on the concentric circles is explained as follows.

Let the number of concentric circles be N and $\mathbf{Q}^{(i)}$ be the set of DWT coefficients on the i-th concentric circle ($1 \leq i \leq N$). Assume that r_i is the radius of the i-th concentric circle, where the circle radius is numbered from small value to large value, i.e., r_1 is the smallest one and r_N is the biggest one. To make a uniform sampling of DWT coefficients, the radius sequence $[r_1, r_2, \ldots, r_N]$ is selected as an arithmetic progress, where the difference d_r between any two neighbor radiuses is determined by the below equation.

$$d_r = \left\lfloor \frac{S}{2N} \right\rfloor \tag{5}$$

where the symbol $\lfloor\cdot\rfloor$ is the downward rounding operation and S is the fixed size of normalized image. Thus, the i-th radius r_i is calculated by the following equation.

$$r_i = i \times d_r \quad (1 \leq i \leq N) \tag{6}$$

Let $q(x, y)$ be the DWT coefficient in the x-th row and y-th column of the LL sub-band. Suppose that (x_c, y_c) is the central coordinates of the LL sub-band. Consequently, $x_c = S/4 + 0.5$ and $y_c = S/4 + 0.5$. Next, the Euclidean distance $d_{x,y}$ from $q(x, y)$ to the center (x_c, y_c) is calculated as follows.

$$d_{x,y} = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{7}$$

Therefore, the DWT coefficients on the i-th concentric circle can be determined by the following rule.

$$\mathbf{Q}^{(i)} = \left\{ q(x,y) \middle| \ \left| d_{x,y} - r_i \right| \leq \Delta d \right\} \tag{8}$$

where Δd is a threshold used for error control. Here we use error threshold because the coordinates are discrete and only few DWT coefficients precisely fall on the circle, i.e., $d_{x,y} = r_i$. For each set $\mathbf{Q}^{(i)}$, its mean m_i and variance v_i are selected as image features, which can be calculated by the below equations.

$$m_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} Q^{(i)}(j) \tag{9}$$

$$v_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} \left(Q^{(i)}(j) - m_i \right)^2 \tag{10}$$

where N_i is the total number of DWT coefficients in $\mathbf{Q}^{(i)}$, and $Q^{(i)}(j)$ is the j-th element of $\mathbf{Q}^{(i)}$ $(1 \leq j \leq N_i)$.

2.3. Data encryption and compression

To ensure security of the proposed hashing, the well-known chaotic map called logistic map is exploited to encrypt image features. The logistic map is defined as follows.

$$x_{i+1} = \mu \times x_i (1 - x_i) \tag{11}$$

where μ is the parameter of logistic map, $0 \leq \mu \leq 4$ and $0 \leq x_i \leq 1$. The initial value x_0 and the parameter μ are viewed as secret keys in this paper. Therefore, we exploit logistic map to generate a sequence with N random numbers, i.e., $[R(1), R(2), \ldots, R(N)]$, under the control of secret keys. Data encryption is conducted by calculating products between image features and their corresponding random number as follows.

$$M_i = m_i \times R(i) \quad (1 \leq i \leq N) \tag{12}$$

$$V_i = v_i \times R(i) \quad (1 \leq i \leq N) \tag{13}$$

Next, data normalization is computed as follows.

$$M_i' = \frac{M_i - u_1}{\delta_1} \tag{14}$$

$$V_i' = \frac{V_i - u_2}{\delta_2} \tag{15}$$

where u_1 and u_2 are the means of $\{M_1, M_2, ..., M_N\}$ and $\{V_1, V_2, ..., V_N\}$, δ_1 and δ_2 are their standard deviations, respectively. Here, M_i' and V_i' are viewed as the coordinates of a point (M_i', V_i') in the 2D plane. Therefore, there are N points in total. The Euclidean distance $U_{i,j}$ between any two points (M_i', V_i') and (M_j', V_j') ($j \neq i$) is computed as follows.

$$U_{i,j} = \sqrt{(M_i' - M_j')^2 + (V_i' - V_j')^2} \quad (i \neq j) \tag{16}$$

Consequently, there are $L = \mathrm{C}_N^2 = N(N-1)/2$ distances. For simplicity, let B_i be the i-th distance ($1 \leq i \leq L$). Thus, it is quantized to make an integer hash element.

$$h(i) = \mathrm{round}(B_i) \tag{17}$$

where round(·) is the rounding operation. Finally, our image hash is obtained by concatenating these hash elements as follows.

$$\mathbf{h} = [h(1), h(2), ..., h(L)] \tag{18}$$

Clearly, our hash length is L integers.

2.4. Pseudo-code

The pseudo-code of our image hashing is illustrated as follows.

```
Algorithm: Image hashing via visual attention model and ring partition
Input: Color image and the normalized image size $S$.
Output: Image hash $\mathbf{h}$.
1. Color image is mapped to a fixed size $S \times S$ by bi-cubic interpolation.
2. The fixed-size color image is converted to YCbCr color space and the luminance
   component $\mathbf{Y}$ is taken for representation.
3. Saliency map of the luminance image $\mathbf{Y}$ is calculated by PFT model.
4. The saliency map is decomposed by a single level 2D DWT and the LL sub-band is taken
   for representation.
5. The LL sub-band is divided into $N$ concentric circles via ring partition. Let $\mathbf{Q}^{(i)}$ be the set
   of DWT coefficients on the $i$-th concentric circle ($1 \leq i \leq N$). Calculate the mean $m_i$ and
   variance $v_i$ of $\mathbf{Q}^{(i)}$.
6. Generate $N$ random numbers by logistic map, i.e., $[R(1), R(2), …, R(N)]$. Perform data
   encryption by Eq. (12) and Eq. (13) and data normalization by Eq. (14) and Eq. (15).
7. Calculate Euclidean distance between any two points by Eq. (16) and quantize every
   distance to make an integer hash element $h(i)$ ($1 \leq i \leq L$) by Eq. (17). Concatenate all hash
   elements and obtain the image hash $\mathbf{h} = [h(1), h(2), ..., h(L)]$.
```

2.5. Hash similarity

Since our image hash is an integer sequence, we select the L_1 norm to measure similarity between two input hashes. Let $\mathbf{h_1} = [h_1(1), h_1(2), \dots, h_1(L)]$ and $\mathbf{h_2} = [h_2(1), h_2(2), \dots, h_2(L)]$ be the hashes of two images. Thus, their L_1 norm is calculated by the following equation.

$$D(\mathbf{h_1}, \mathbf{h_2}) = \sum_{i=1}^{L} |h_1(i) - h_2(i)| \tag{19}$$

in which $h_1(i)$ and $h_2(i)$ are the i-th elements of $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. In general, a bigger L_1 norm means lower similarity of two images of input hashes. Therefore, if the L_1 norm D is bigger than a threshold, the images of input hashes are viewed as different images. Otherwise, they are a pair of similar images.

3. Experimental results

In the following experiments, the parameters of our image hashing are set as follows. Input image are converted to a fixed size 512 × 512, the used number of concentric circles is 18, the error threshold is 3.5, the parameter of logistic map is 4 and the initial x value is 0.4. In other words, S = 512, N = 18, Δd = 3.5, $\mu = 4$ and x_0 = 0.4. Therefore, our hash length is $L = \mathrm{C}_{18}^{2} = 153$ integers. Our performances of robustness, discrimination, security, hash storage and effect of the number of concentric circles are discussed in Sections 3.1~3.5. Section 3.6 presents performance comparisons.

3.1. Robustness

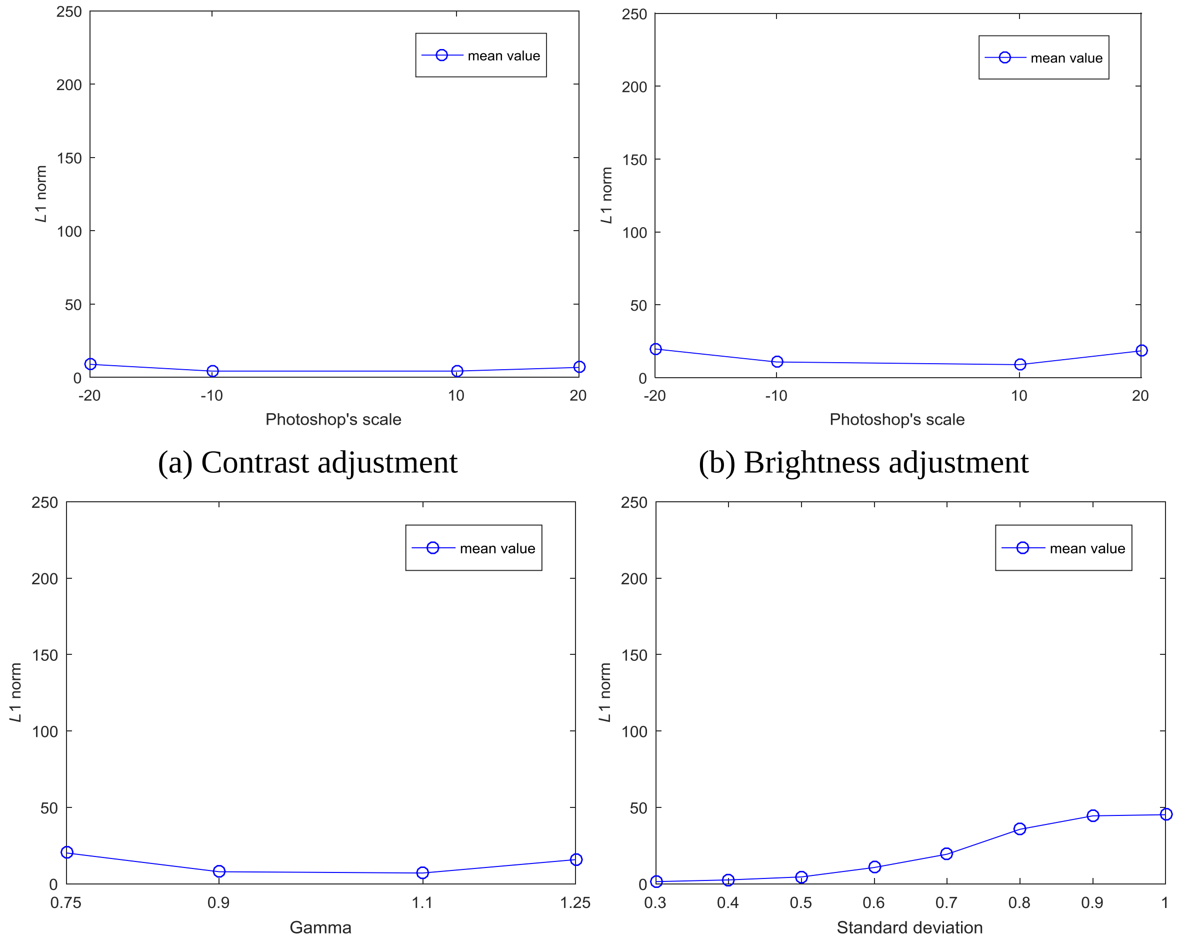
The 24 color images in the open dataset called Kodak image database [30] are selected as test images. To produce similar versions of these 24 color images, the image processing tools called Photoshop, MATLAB and StirMark 4.0 [31] are exploited to conduct robustness attacks. Photoshop provides the adjustments of contrast and brightness (each operation uses 4 parameters). MATLAB provides gamma correction (4 parameters are used), 3 × 3 Gaussian low-pass filtering (8 parameters are used), salt & pepper noise (10 parameters are used) and speckle noise (10 parameters are used). StirMark provides JPEG compression (8 parameters are used), watermark embedding (10 parameters are used), image scaling (6 parameters are used) and combinational attack of rotation and cropping (16 parameters are used). For the combinational attack, input image is first rotated, and the central part of the rotated image sized 361 × 361 is taken for hash generation. Therefore, every color image has 80 similar versions and thus the total pairs of similar images are 24 × 80 = 1920.

To measure visual similarity of these image pairs, the well-known image quality assessment metric called structural similarity index measurement (SSIM) [32] is used. Note that the maximum value of SSIM is 1. In general, the bigger the SSIM value, the more similar the image pairs. Table 1 presents SSIM results under different robustness attacks. It is found that the average SSIM values are bigger than 0.8 and all standard deviations are small. This illustrates that these attacked images are similar with their original images. Since SSIM cannot be applied to rotated images and scaled images, the SSIM results of image scaling and combinational attack are not listed in Table 1. Figure 3 illustrates our robustness performances under different operations, where the x-axis is the parameter of the used operation and the y-axis is the mean value of the L_1 norms of 24 pairs of similar images.

From Figure 3, it is observed that most of mean values are smaller than 50, except a few cases in Figures 3 (e), (g), (i) and (j). If the threshold is selected as 70, our image hashing can correctly recognize 92.19% similar images, indicating good robustness performance.

Table 1. SSIM results under robustness attacks.

Robustness attack	Average SSIM	Standard deviation
Brightness adjustment	0.9914	0.0043
Contrast adjustment	0.9974	0.0014
Gamma correction	0.9778	0.0185
3 × 3 Gaussian low-pass filtering	0.9478	0.0474
Speckle noise	0.8465	0.0979
Salt and pepper noise	0.8819	0.0655
JPEG compression	0.9222	0.0403
Watermark embedding	0.9821	0.0175

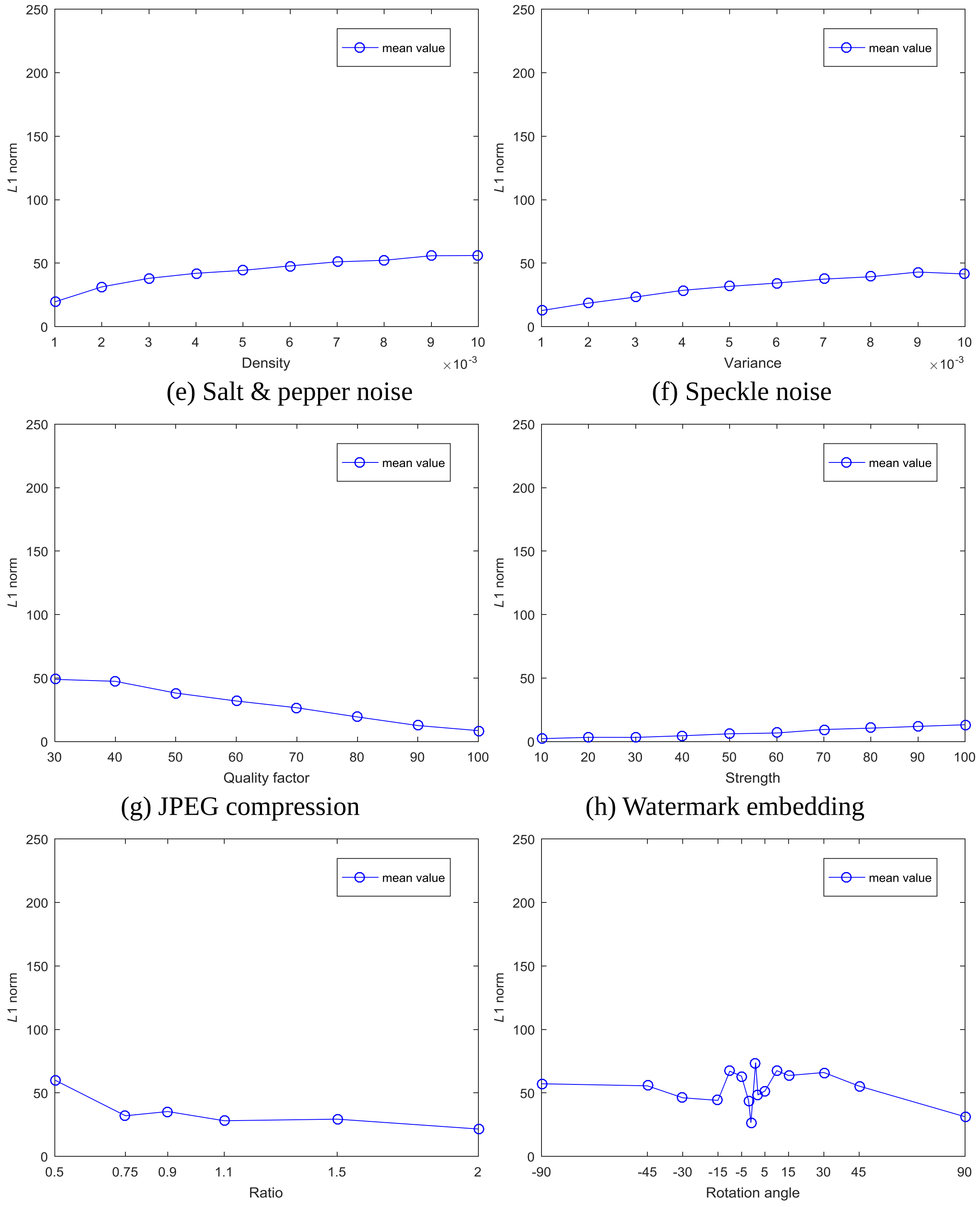


(a) Contrast adjustment

(b) Brightness adjustment

(c) Gamma correction

(d) 3 × 3 Gaussian low-pass filtering



(e) Salt & pepper noise

(f) Speckle noise

(g) JPEG compression

(h) Watermark embedding

(i) Image scaling

(j) Rotation and cropping

Figure 3. Robustness test based on 24 color images.

3.2. Discrimination

The image dataset used in [12] is selected as database for discrimination test. This dataset contains 200 color images, where 67 images are collected via Internet, 33 images are generated by cameras and 100 images are picked out from the Ground Truth Database [33]. These images contain various contents, such as sport, building, scenery and human being, and their sizes range from to 256 × 256 to 2048 × 1536. For each test image, we calculate L_1 norms between its hash and the hashes of other 199 images. Therefore, the total number of the valid L_1 norms is 200 × 199/2 = 19900. Figure 4 presents data distribution of these 19900 distances, where the *x*-axis is the L_1 norm and the *y*-axis is its frequency. It is found that the maximum and minimum L_1 norms are about 305 and 46, the mean L_1 norm is 183.91 and the standard deviation is 39.05. From Figure 4, it can be seen that L_1 norm between hashes of different images is large enough. If the threshold is 70, our image hashing will only mistakenly judge 0.06% different images as similar images. A low false detection rate validates good discrimination of our hashing. In fact, our discrimination and robustness are both closely related to the selected threshold. Generally, a small threshold can improve discrimination, but it will hurt robustness. Table 2 illustrates our detection performances under different thresholds, where robustness performance is described by the percent of similar images judged as similar images and the discrimination is measured by the percent of different images detected as similar images. In practice, we can select a proper threshold to make a tradeoff between robustness and discrimination according to the requirement of specific application.

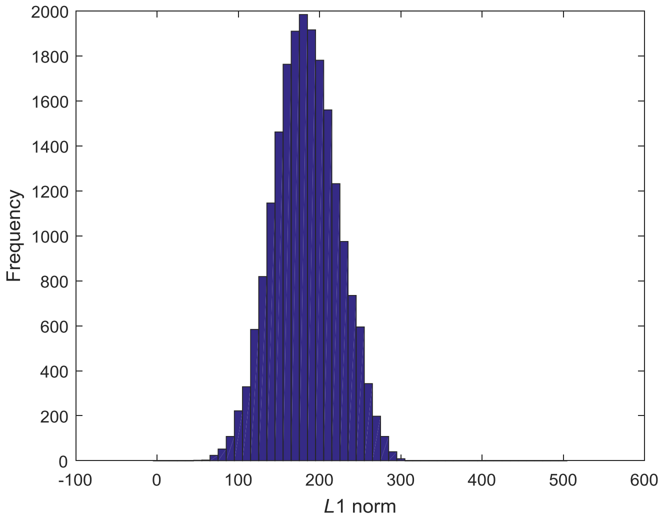


Figure 4. Discrimination test based on 200 color images.

Table 2. Our detection performances under different thresholds.

Threshold	Similar images judged as similar images	Different images detected as similar images
60	87.40%	0.01%
70	92.19%	0.06%
80	95.78%	0.24%
90	97.55%	0.60%
100	98.39%	1.41%
110	98.80%	2.78%
120	99.48%	4.93%
130	99.84%	8.57%
140	100%	13.48%

3.3. Security

The benchmark color image Baboon is selected as test image to test security of our image hashing. To do this, a group of secret keys (i.e., μ and initial x value of logistic map) is firstly used to control hash generation of Baboon. Then, another 100 different groups of secret keys are exploited to extract image hashes of Baboon. Finally, L_1 norms between the first hash and other 100 hashes are calculated. Figure 5 is the L_1 norm results, where the x-axis is the index of wrong keys and the y-axis is the L_1 norm. It is observed that the maximum and minimum L_1 norms are 230 and 60, the mean and standard deviation are 152.32 and 32.49, respectively. The mean L_1 norm is much bigger than most mean values of similar images. This illustrates good security of our algorithm.

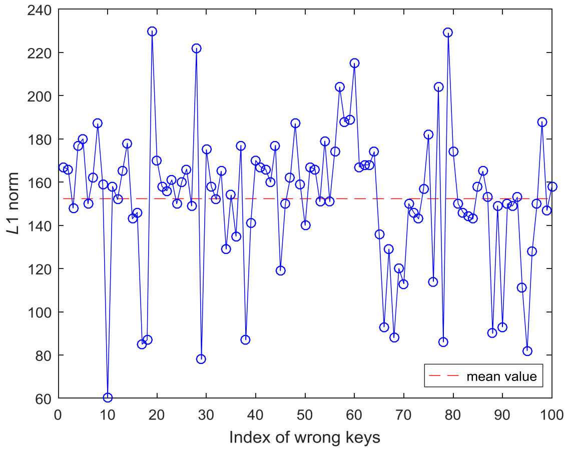


Figure 5. L_1 norm between hashes of an image generated with wrong keys.

3.4. Hash storage

To view the required bits of storing our image hash, the hashes generated in discrimination test are selected as data source for analysis. As every hash contains 153 elements, the total number of hash elements in the data source is 200 × 153 = 30600. Figure 6 is the distribution of these hash elements, where the x-axis is the element value and the y-axis is its frequency. Clearly, the minimum element value is 0 and the maximum element value is 6. Since 3 bits can represent an integer ranging from 0 to $2^3 - 1 = 7$, storage of a hash element only needs 3 bits. Therefore, storage cost of our image hash is $153 \times 3 = 459$ bits.

3.5. Effect of the number of concentric circles

Our hash length and classification performance are closely related to the number of concentric circles, i.e., the N value. To view the effect of N value, we select five values (i.e., N =10, N =12, N =16, N =18 and N = 20) to calculate hashes of similar images and different images used in the Sections 3.1 and 3.2, respectively. In the experiments, only the N value is different and other parameters are not changed. To compare classification performances, the receiver operating characteristics (ROC) graph [34] is utilized to conduct quantitative analysis. In the ROC graph, the x-axis and the y-axis are generally defined as FPR (false positive rate) and TPR (true positive rate), respectively. The FPR and TPR can be calculated by the below equations.

$$\mathrm{FPR} = \frac{\text{Number of different images detected as similar images}}{\text{Total number of different images}} \tag{20}$$

$$\mathrm{TPR} = \frac{\text{Number of similar images judged as similar images}}{\text{Total number of similar images}} \tag{21}$$

It is clear that FPR and TPR are the indicators of discrimination and robustness, respectively. A low FPR means good discrimination and a big TPR implies good robustness. Note that ROC curve is plotted with some points (FPR, TPR). Therefore, the curve close to the top-left corner (a low FPR and a big TPR) shows better classification performance than that far away from it.

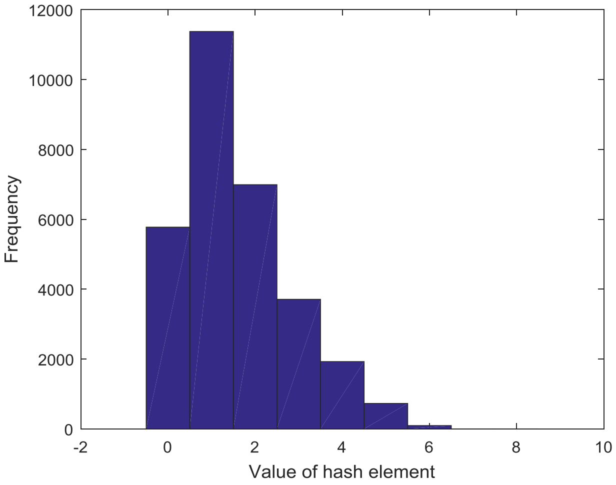


Figure 6. Distribution of the elements of 200 image hashes.

Figure 7 presents the ROC curves under different *N* values, where the curves near the top-left corner are zoomed in. From Figure 7, it can be seen that the ROC curves of *N* = 18 and *N* = 20 are closer to the top-left corner than those of other *N* values. Therefore, it is intuitively found that the classification performances of *N* = 18 and *N* = 20 are better than those of other values. To make quantitative comparisons, area under the ROC curve (AUC) [34] is selected as the metric, whose value ranges from 0 to 1. In general, a bigger AUC means better classification performance. It is found that the AUCs of *N* = 10, *N* = 12, *N* = 16, *N* = 18 and *N* = 20 are 0.99616, 0.99717, 0.99843, 0.99903 and 0.99902, respectively. Obviously, the AUC of *N*=18 is the biggest one. It means that our classification performance with *N*=18 is better than those with other *N* values. Our hashing reaches desirable classification performance when *N*=18. This is because: (1) A small *N* value (e.g., 10 and 12) means few features in the hash and then the discrimination is hurt. (2) A big *N* value (e.g., 20) will lead to overlapping regions between neighbor concentric circles and features extracted from these overlapping regions are not helpful to discrimination but decrease robustness. Moreover, the bigger the *N* value, the longer the hash length. A moderate *N* value can make a reasonable length. Table 3 lists the hash lengths and AUCs of different *N* values.

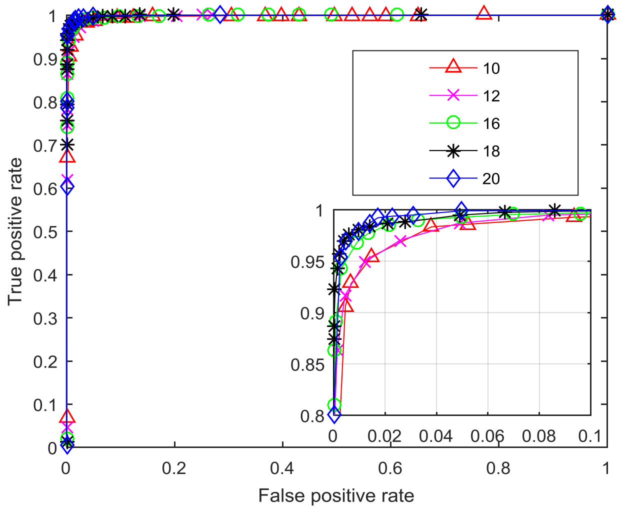


Figure 7. ROC curves under different numbers of concentric circles.

Table 3. Our performances under different numbers of concentric circles.

N	AUC	Hash length (integer)
10	0.99616	45
12	0.99717	66
16	0.99843	120
18	0.99903	153
20	0.99902	190

3.6. Performance comparisons

To demonstrate advantages, we compare our image hashing with five well-known hashing algorithms. These compared hashing algorithms are SVD-CSLBP hashing [19], random-walk hashing [20], CVA-DWT hashing [12], RT-DCT hashing [8], and hybrid features-based hashing [22]. During the comparisons, all images are resized to 512 × 512 for hash generation, other parameters of the compared algorithms are used as their reported settings, and their reported similarity metrics are also taken. Therefore, the hash lengths of random-walk hashing, CVA-DWT hashing and RT-DCT hashing are 144, 960 and 240 bits, respectively. The lengths of SVD-CSLBP hashing and hybrid features-based hashing are 64 and 104 floats, respectively. According to the IEEE standard [35], 32 bits are needed for storage of a float. Therefore, the lengths of SVD-CSLBP hashing and hybrid features-based hashing are 2048 and 3328 bits, respectively. For our image hashing, the experimental results with N = 18 are taken for comparisons. Thus, our hash length is 153 integers equivalent to 459 bits in binary form.

Figure 8 presents ROC curve comparisons between our hashing and the compared algorithms. From Figure 8, it is observed that our curve is closer to the top-left corner than the curves of the compared algorithms. Therefore, it can intuitively draw a conclusion that the classification performance of our hashing is better than those of the compared algorithms. The AUCs of all algorithms are also calculated, and the results of our hashing, SVD-CSLBP hashing, random-walk hashing, CVA-DWT hashing, RT-DCT hashing and hybrid features-based hashing are 0.99903, 0.82011, 0.95793, 0.98033, 0.83674 and 0.97044, respectively. Our AUC is the biggest one among the results of all algorithms. This validates that our hashing shows better classification performance than the compared algorithms. Computational time of these algorithms is also compared. To do this, the total consumed time of extracting hashes in the respective discrimination test are calculated to find the average time for producing a hash. All algorithms are coded with MATLAB and run on a PC with the configuration that the CPU is the Intel Core i7-7700 CPU with 3.6 GHz, and the capacity of RAM is 8 GB. It is found that the average time of our hashing, SVD-CSLBP hashing, random-walk hashing, CVA-DWT hashing, RT-DCT hashing and hybrid features-based hashing are 0.064, 0.101, 0.031, 0.036, 1.349 and 31.87 seconds, respectively. Table 4 summarizes performance comparisons between our hashing and the compared algorithms. Our hashing has the best classification performance in terms of AUC. Our hashing runs faster than SVD-CSLBP hashing, RT-DCT hashing and hybrid features-based hashing, but it is slower than random-walk hashing and CVA-DWT hashing. For hash length, our length is much shorter than those of SVD-CSLBP hashing, CVA-DWT hashing and hybrid features-based hashing, but it is longer than those of random-walk hashing and RT-DCT hashing.

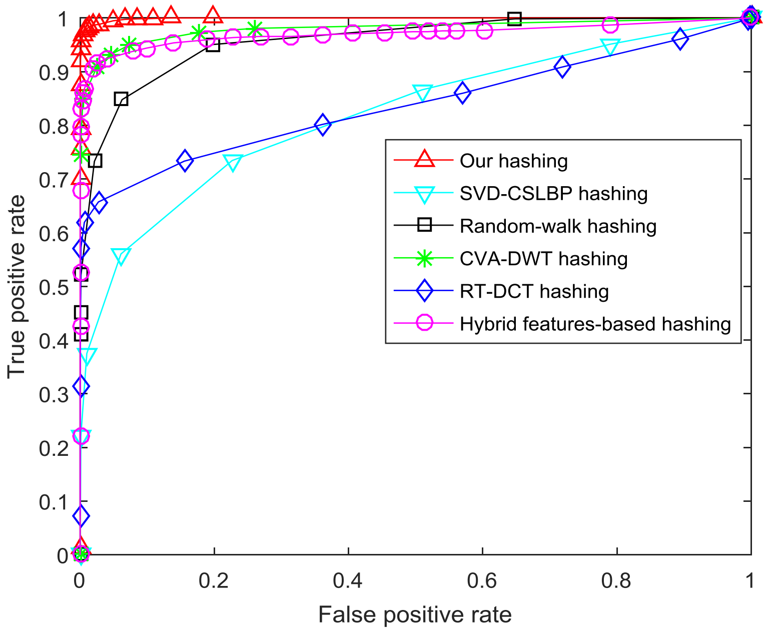


Figure 8. Classification performance among different algorithms in terms of ROC curves.

Table 4. Performance comparisons among different algorithms.

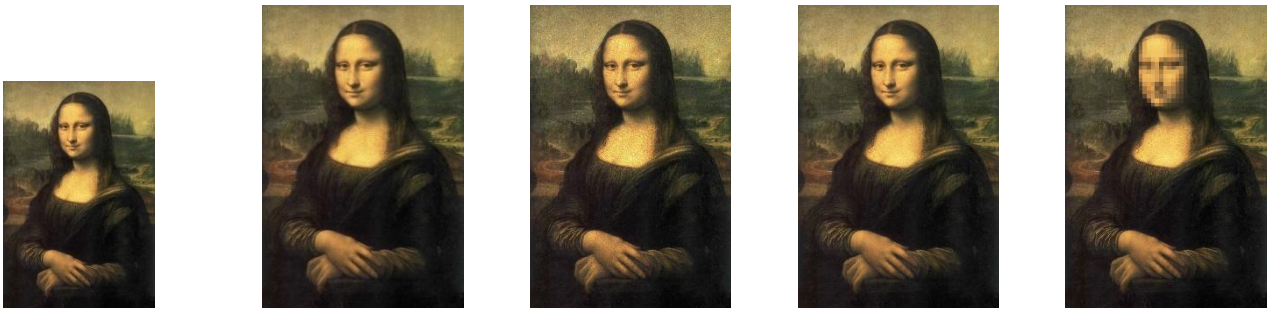
Algorithm	AUC	Time (s)	Hash length (bit)
Our hashing	0.99903	0.064	459
SVD-CSLBP hashing	0.82011	0.101	2048
Random-walk hashing	0.95793	0.031	144
CVA-DWT hashing	0.98033	0.036	960
RT-DCT hashing	0.83674	1.349	240
Hybrid features-based hashing	0.97044	31.87	3328

4. Application to image copy detection

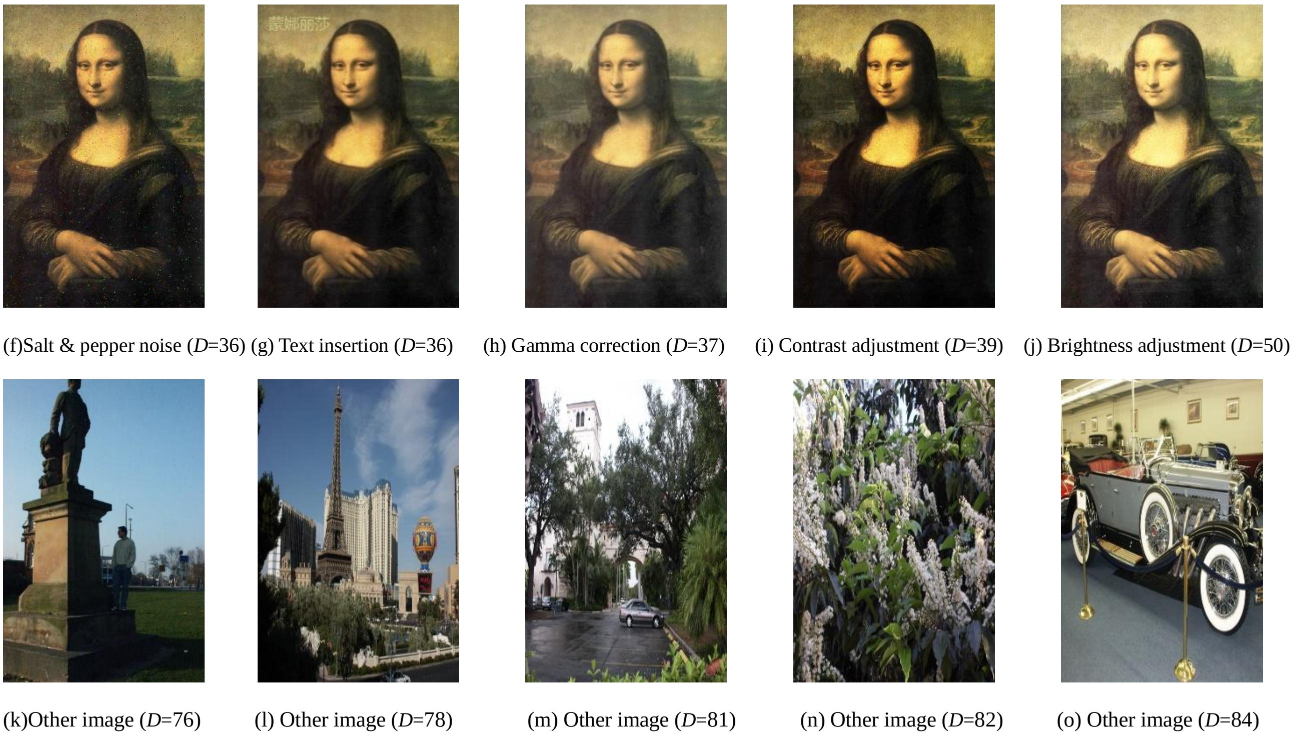
To test the performance of our hashing in application to image copy detection, the well-known painting called Mona Lisa created by the Italian artist named Leonardo Da Vinci is selected as test image. This test image is a color image sized 1976 × 2976, as shown in Figure 9. Ten digital operations are chosen to produce image copies of the test image. The used operations include JPEG compression (QF = 70), contrast adjustment (Photoshop's scale is 60), brightness adjustment (Photoshop's scale is 60), image scaling (ratio is 0.75), mosaic processing (cell size is 40), text insertion (font size is 150), gamma correction (gamma is 0.75), 3 × 3 Gaussian low-pass filtering (standard deviation is 1), speckle noise (variance is 0.01) and salt & pepper noise (density is 0.01). To construct a large dataset for retrieval, the open image database called UCID [36] is used in the experiment. This database has 1338 different color images, whose sizes are 384 × 512 or 512 × 384. Consequently, a test dataset with 1348 color images is obtained by adding the above 10 image copies to the UCID.

Figure 9. Test image of Mona Lisa.

In this experiment, the hash of the test image is first calculated, the hashes of 1348 images in the test dataset are then extracted, and finally the L_1 norms between the hash of the test image and the 1348 hashes are computed. To view the detected results, the 1348 L_1 norms are sorted in ascending order. Figure 10 presents the top 15 returned images. It can be seen that the top 10 images are the copies of the test image. In addition, the L_1 norms between hashes of the test image and other images are much bigger than the L_1 norms between hashes of the test image and its copies. This validates effectiveness of our hashing in application to copy detection.



(a) Scaling (*D*=5) (b) JPEG compression (*D*=9) (c) Speckle noise (*D*=11) (d) low-pass filtering (*D*=11) (e) Mosaic (*D*=21)



(f)Salt & pepper noise (*D*=36) (g) Text insertion (*D*=36) (h) Gamma correction (*D*=37) (i) Contrast adjustment (*D*=39) (j) Brightness adjustment (*D*=50)

(k)Other image (*D*=76) (l) Other image (*D*=78) (m) Other image (*D*=81) (n) Other image (*D*=82) (o) Other image (*D*=84)

Figure 10. The returned images sorted in ascending order according to the L_1 norm D.

5. Conclusions

In this paper, we have proposed a novel robust image hashing based on PFT model and ring partition. Since visual attention model can find salient regions of attention focus, the use of PFT model helps to improve perceptual robustness of our proposed hashing. As ring partition can divide image into concentric circles invariant to rotation, our hash construction with concentric circles is robust to image rotation. Various experiments have been conducted to test the performances of our hashing. The results have demonstrated that our hashing is robust to digital operations, reaches good discrimination and can be applied to image copy detection. Comparisons with some well-known hashing algorithms have shown that our hashing has better classification performance than the compared algorithms and reaches moderate performances in hash length and computational time.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (61562007, 61762017), Guangxi “Bagui Scholar” Team for Innovation and Research, the Guangxi Natural Science Foundation (2017GXNSFAA198222), the Project of Guangxi Science and Technology (GuiKeAD17195062, GuiKeAD16380008), the Guangxi 1000-Plan of Training Middle-aged/Young Teachers in Higher Education Institutions, Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the Innovation Project of Guangxi Graduate Education (XYCSZ2019066).

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. Many thanks to Miss M. Sun [22] for sharing their code for comparison.

Conflict of interest

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

References

1. Z. Tang, Z. Huang, X. Q. Zhang, et al., Robust image hashing with multidimensional scaling, *Signal Process.*, **137**(2017), 240–250.
2. C. Qin, X. Chen, D. Ye, et al., A novel image hashing scheme with perceptual robustness using block truncation coding, *Inform. Sci.*, **361**(2016), 84–99.
3. Z. Tang, L. Chen, X. Q. Zhang, et al., Robust image hashing with tensor decomposition, *IEEE Trans. Knowl. Data En.*, **31**(2019), 549–560.
4. Z. Tang, S. Wang, X. P. Zhang, et al., Lexicographical framework for image hashing with implementation based on DCT and NMF, *Multimed. Tools Appl.*, **52**(2011), 325–345.
5. F. Lefebvre, B. Macq and J. D. Legat, RASH: Radon soft hash algorithm, In: *Proc. of European Signal Processing Conference*, Toulouse, France, Sep. 3−6, 2002, pp.299–302.
6. A. Swaminathan, Y. Mao and M. Wu, Robust and secure image hashing, *IEEE Trans. Inf. Foren. Secur.*, **1**(2006), 215–230.
7. V. Monga and B. L. Evans, Perceptual image hashing via feature points: performance evaluation and trade-offs, *IEEE Trans. Image Process.*, **15**(2006), 3453–3466.
8. Y. Ou and K. H. Rhee, A key-dependent secure image hashing scheme by using Radon transform, In: *Proc. of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, pp.595–598, 2009.
9. Z. Tang, S. Wang, X. P. Zhang, et al., Structural feature-based image hashing and similarity metric for tampering detection, *Fundam. Inf.*, **106**(2011), 75–91.
10. C. Qin, C. C. Chang and P. L. Tsou, Robust image hashing using non-uniform sampling in discrete Fourier domain, *Digit. Signal Process.*, **23**(2013), 578–585.
11. Z. Tang, X. Q. Zhang, L. Huang, et al., Robust image hashing using ring-based entropies, *Signal Process.*, **93**(2013), 2061–2069.
12. Z. Tang, Y. Dai, X. Q. Zhang, et al., Robust image hashing via colour vector angles and discrete wavelet transform, *IET Image Process.*, **8**(2014), 142–149.
13. L. Ghouti, Robust perceptual color image hashing using quaternion singular value decomposition, In: *Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing* (ICASSP 2014), pp.3794–3798, 2014.
14. C. Yan, C. Pun and X. Yuan, Quaternion-based image hashing for adaptive tampering localization, *IEEE Trans. Inf. Foren. Secur.*, **11**(2016), 2664–2677.
15. C. Qin, X. Chen, J. Dong, et al., Perceptual image hashing with selective sampling for salient structure features, *Displays*, **45**(2016), 26–37.
16. Z. Tang, X. Q. Zhang, X. Li, et al., Robust image hashing with ring partition and invariant vector distance, *IEEE Trans. Inf. Foren. Secur.*, **11**(2016), 200–214.
17. R. K. Karsh, R. H. Laskar and B. B. Richhariya, Robust image hashing using ring partition-PGNMF and local features, *SpringerPlus*, **5**(2016), 1–20.
18. R. K. Karsh, R. H. Laskar and Aditi, Robust image hashing through DWT-SVD and spectral residual method, *EURASIP J. Image Vide.*, **2017**(2017), 1–17.

19. R. Davarzani, S. Mozaffariand and K. Yaghmaie, Perceptual image hashing using center-symmetric local binary patterns, *Multimed. Tools Appl.*, **75**(2016), 4639–4667.
20. X. Huang, X. Liu, G. Wang, et al., A robust image hashing with enhanced randomness by using random walk on zigzag blocking, In: *Proc. IEEE Trustcom/BigDataSE/ISPA*, pp.23–26, 2016.
21. R. K. Karsh, A. Saikia and R. H. Laskar, Image authentication based on robust image hashing with geometric correction, *Multimed. Tools Appl.*, **77**(2018), 25409–25429.
22. C. Qin, M. Sun and C.-C. Chang, Perceptual hashing for color images based on hybrid extraction of structural features, *Signal Process.*, **142**(2018), 194–205.
23. Z. Tang, Z. Huang, H. Yao, et al., Perceptual image hashing with weighted DWT features for reduced-reference image quality assessment, *Comput. J.*, **61** (2018), 1695–1709.
24. L. Itti, C. Koch and E. Niebur, A model of saliency based visual attention for rapid scene analysis, *IEEE Trans. Patt. Anal. Mac. Intell.*, **20**(1998), 1254–1259.
25. D. Walther and C. Koch, Modeling attention to salient proto-objects, *Neural Networks*, **19**(2006), 1395–1407.
26. X. Hou and L. Zhang, Saliency detection: A spectral residual approach, In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp.1–8, 2007.
27. C. Guo, Q. Ma and L. Zhang, Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform, In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp.1–8, 2008.
28. Z. Tang, X. Q. Zhang and S. Zhang, Robust perceptual image hashing based on ring partition and NMF, *IEEE Trans. Knowl. Data En.*, **26**(2014), 711–724.
29. Z. Tang, L. Huang, X. Q. Zhang, et al., Robust image hashing based on color vector angle and canny operator, *AEÜ-Int. J. Electron. Commun.*, **70**(2016), 833–841.
30. Kodak Lossless True Color Image Suite. Available online: http://r0k.us/graphics/kodak/.
31. F. A. P. Petitcolas, Watermarking schemes evaluation, *IEEE Signal Process. Mag.*, **17**(2000), 58–64.
32. Z. Wang, A. C. Bovik, H. R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, **13**(2004), 600–612.
33. Ground Truth Database. Available online: http://www.cs.washington.edu/research/imagedatabase/groundtruth/.
34. T. Fawcett, An introduction to ROC analysis, *Patt. Recog. Lett.*, **27**(2006), 861–874.
35. IEEE Std754–2008, IEEE Standard for Floating-Point Arithmetic, pp.1–70, 2008.
36. G. Schaefer and M. Stich, UCID-an uncompressed color image database, *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pp.472–480, 2004.