Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7691091, 16 pages
https://doi.org/10.1155/2022/7691091

*Research Article*

# Image Perceptual Hashing for Content Authentication Based on Geometric Invariant Vector Distance

**Huifen Xing**,[1] **Shuchao Wang**,[2] **Qilin Wu**,[1] **and Honghai Wang**[1]

[1]*School of Computer and Artificial Intelligence, Chaohu University, Hefei 238024, China*
[2]*School of Mathematics and Big Data, Chaohu University, Hefei 238024, China*

Correspondence should be addressed to Huifen Xing; xhf870426@126.com

Received 3 October 2022; Revised 24 November 2022; Accepted 6 December 2022; Published 24 December 2022

Academic Editor: Shangce Gao



Image perceptual hashing is broadly applied in image content authentication, recognition, retrieval, and social media hotspot event detection. An image authentication algorithm is put forward based on the Itti visual saliency model and geometric invariant vector distance. To begin with, the image is preprocessed and weighted by the Itti model and contourlet transform. After that, the weighted image is randomly divided into blocks, and the image feature vector is constructed by calculating the geometric invariant vector distance on both Hu invariant moment vector and maximum singular value vector of the random blocks. In the end, the feature vector is quantized and encrypted to generate the ultimate hash. Experimental results illustrate that when the threshold $T = 70$, the true positive rate $P_{\mathrm{TPR}}$ for duplicate images stands at 0.96574, while the false rate $P_{\mathrm{FPR}}$ of different images is merely 0.0224, with the total error rate reaching the minimum value (0.0566). Furthermore, the AUC value of the proposed algorithm is 0.9951, which is higher than that of the comparison algorithms, indicating that the algorithm has better performance than other state-of-the-art algorithms in terms of various visual content-preserving attacks.

## 1. Introduction

As a wide variety of image editing tools are getting developed and popularized, such as ACDSee, Corel Paint Shop Pro Photo, and Photoshop, malicious users can easily modify, copy, edit, and tamper images, resulting in numerous duplicate images growing exponentially on the networks, which brings serious challenges to the copyright protection and content authentication of images [1–3]. There are some content-preserving operations which usually refer to those images that undergo various kinds of image modifications such as rescaling, occlusion, noise adding, and luminance and color change. Therefore, content-based image copy detection has been dealt with as speedily as possible. The purpose of content-based image detection is to find the duplicate version of a given copyright image in the database. Duplicate images and similar images are visually very similar to copyright images. Figure 1 shows similar images and duplicate images of a copyright image. Therefore, how to effectively distinguish similar images from duplicate images is a difficult problem in image copy detection. This study focuses on how to use image hashing technology to effectively detect content-based copy images.

At present, there are two kinds of effective image content detection methods. The first is digital watermarking technology [4], but the robustness of watermarking against various content-preserving attacks limits the practical application. The other is image hashing technology [5]. Compared to the former, image hashing only depends on the image content itself and does not need to damage the image quality. The image hashing generates a brief summary through the extraction and compression of image perceived content and information, which is used for recording or representing the image content or further used for finding the location where the image content is tampered [6, 7]. It provides a high-efficiency technology for image content authentication. Because image hashing can effectively reduce the cost of image storage and computational complexity, it realizes the quick image processing. Also, the hashing has been applied in image retrieval [8] and other applications [9–13].

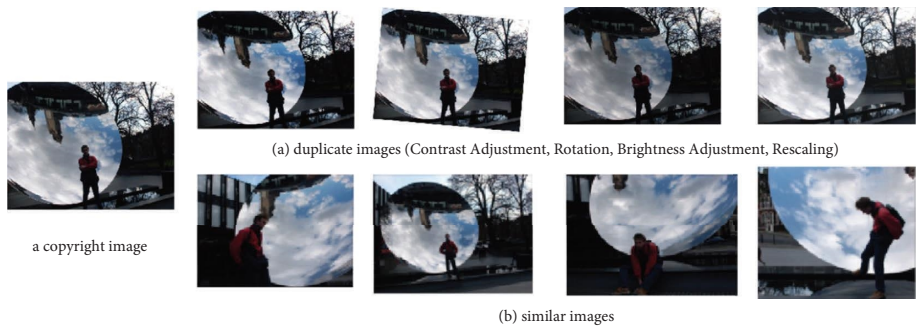



Figure 1: Similar and duplicate images of a copyright image.

In recent years, the image hashing algorithms [14–20] ignore the impact of human visual model on hashing perceptual characteristics and pay less attention to the impact of geometric invariant vector distance on image hash robustness. However, in reference [21], Liu and Huang presented an effective approach based on geometric invariant vector distance, that is, to construct image hashing by the vector distance between Hu invariant moment vectors in the spatial domain and DCT (discrete cosine transform) coefficients in the frequency domain. Although this algorithm focused on the geometric invariance and had good classification performance in image copy detection and geometric attack, it did not fully consider the perceptual characteristics of the human visual model. In reference [22], an image hashing algorithm was introduced using the Itti visual saliency model and DWT (discrete wavelet transformation), which had good robustness and uniqueness, but DWT was not as good as contourlet transform (CT) in sparse expression of images. In view of these, this study chooses the Itti visual saliency model, contourlet transform, and geometric invariant vector distance to study the image hashing algorithm. Compared with previous algorithms, the main contributions of this hashing algorithm are as follows:

(1) A weighted image representation is proposed by making use of the Itti visual saliency model and contourlet transform. The Itti visual saliency model can efficiently capture the visual saliency region of the image and can make a compromise between the calculation speed and the saliency detection performance. Compared with wavelet transform, contourlet transform has good multiresolution, localization, and directivity, and the coefficient energy is more concentrated, which can better sparse represent the image. Weighting the above two factors to generate a weighted image and extracting hashing features from it can ensure that the algorithm achieves a good balance between perceptual robustness and discrimination.

(2) The geometric invariant vector distances on the Hu invariant moment vector and the maximum singular value vector are extracted to construct the hash. Hu invariant moment and maximum singular value vector have strong stability against various attacks. At the same time, geometric invariant vector distance is not only robust to geometric transformation but also effectively distinguishes different images, so that the algorithm has good classification performance.

(3) Two keys ensure the security of the algorithm. First, the image is randomly partitioned by key $K_1$ and then the hash eigenvalue is chaotically encrypted by key $K_2$.

Extensive experiments are done to verify the performances of our hashing algorithm. The results illustrate that when the threshold $T = 70$, the true positive rate $P_{\mathrm{TPR}}$ for duplicate images stands at 0.96574, while the false rate $P_{\mathrm{FPR}}$ of different images is merely 0.0224, with the total error rate reaching the minimum value (0.0566). Furthermore, the AUC value of the proposed algorithm is 0.9951, which is higher than that of the comparison algorithms, indicating that the algorithm has better performance than other state-of-the-art algorithms in terms of various visual content-preserving attacks. The rest of this study is organized as follows. Section 2 describes the related work, Section 3 describes the steps of the proposed algorithm, Section 4 analyzes experimental results, and Section 5 concludes this study.

## 2. Related Work

At present, image hashing algorithms can be roughly divided into two categories: signal processing and machine learning. Generally, machine learning is mainly used in image retrieval, while signal processing is mainly applied in image authentication, image quality evaluation, and copy detection [7]. Since this study focuses on the hashing based image copy detection, it puts emphasis on the hashing algorithms about the signal processing.

*2.1. Image Hashing Algorithms Based on Signal Processing.* The scheme [14] distilled image boundary information by utilizing the Canny operator and applied two-dimensional DWT for edge detection. The hash sequence was obtained by weighting the DWT coefficients of different wavelet subbands, with good property in the quality evaluation of semireference images. Abdullahi et al. proposed an algorithm based on FMT (Fourier–Mellin Transform) and fractal encoding, which is resistant to most content-invariant

operations and runs faster in reference [15]. The scheme [16] used shape context and SURF to generate a faster hash. The scheme [17] combined Kaze features and SVD (singular value decomposition) to generate hash sequences, which were robust to gamma correction. The scheme [18] took advantage of quaternion Zernike moments to engender the hash, with good sorting performance. Huang and Liu [19], to construct hash, distilled the biggest gradient magnitude and correlative orientation information from the three color channels and the generated image gradient field. Su et al. integrated the distance metric learning technique and the quantization strategy into the processing of image authentication, which formed a personal authentication framework in reference [20]. Shen and Zhao extracted hashes from the secondary image and quadtree structured features by concatenating color elements COC in reference [23]. The scheme [24] drew steady and distinctive image features from the CIELAB color space to generate robust hash with SVD. These image hashing algorithms ignore the impact of human visual model on hashing perceptual characteristics.

*2.2. Image Hashing Algorithms Based on Machine Learning.* Tang et al. [25] raised an image hashing approach based on DCT and LLE (locally linear embedding), which has good robustness to image scaling and watermark embedding, while the calculation speed of the LLE algorithm is relatively slow. Liang et al. [26] proposed a fast image hashing algorithm based on LC (luminance contrast) and 2D PCA (principal components analysis), which conduced to a low calculating expenditure. Recently, some scholars have proposed some image authentication algorithms based on deep learning hashing algorithms. For example, an integrated feature matching scheme was proposed in Reference [3], which integrates the matching of SIFT (scale-invariant feature transform) features and CNN (convolutional neural networks) features between images for partial-duplicate image detection. Since both the good robustness of SIFT features and the high discriminative power of CNN features are sufficiently explored, the scheme could attain an accurate detection. Li et al. [27] raised an image hashing approach based on neural network, which was the first time that deep neural network was applied to this field. By comparison, this hash algorithm can achieve better performance. Reference [28] presented an image hashing algorithm using multiconstrained CNN, which especially displayed good performance in tampering image detection. Although learning-based hashing methods have achieved satisfactory results in image semantic retrieval, they are not very suitable for content-based image copy detection.

In order to more prominently represent the salient features of the human visual system and improve the robustness and geometric invariance of hashing, this paper chooses the Itti visual saliency model, contourlet transform, and geometric invariant distance to study image hashing algorithms.

## 3. Proposed Algorithm

In this paper, image hashing is studied though making use of techniques such as the Itti visual model, contourlet transform, SVD, and geometric invariant vector distance, in which the Itti visual model can efficiently detect the visual salient area of the image, the Hu invariant moment can resist common geometric attacks, and the maximum singular value vector is robust to various attacks. Firstly, the image is weighted with the Itti visual model and contourlet transform, and secondly, the geometric invariant vector distance of Hu invariant moment vector and maximum singular value vector is constructed as the hashing. Therefore, a better compromise between robustness and discrimination of the perceptual hashing algorithm can be achieved. The framework of hashing generation is displayed in Figure 2.

### 3.1. Algorithm Steps

*3.1.1. Preprocessing.* To begin with, the original image $I_0(x, y)$ is transformed to a normalized image sized $n \times n$ by the bicubic linear interpolation, which is primarily to ensure that images of different sizes have the same length of hash. After that, the Gaussian low-pass filter is utilized for the normalized image by a convolution mask in order for reducing an effect on white noise, interpolation error, and speckle noise on the hash. Generally, filtering can be implemented using a convolution mask. Let $T_{\text{Gaussian}}(i, j)$ be the element in the $i$th row and $j$th column of the convolution mask. Thus, the convolution template can be computed by the following formula:

$$T_{\text{Gaussian}}(i, j) = \frac{T^{(1)}(i, j)}{\sum_i \sum_j T^{(1)}(i, j)}, \tag{1}$$

where $T^1(i, j)$ is defined as

$$T^{(1)}(i, j) = \frac{1}{2\pi\sigma^2} e^{-(i^2+j^2)/2\sigma^2}. \tag{2}$$

In equation (2), $\sigma_\tau$ is the standard deviation of all elements of the convolution template. The image after preprocessing is marked as $I_1(x, y)$.

*3.1.2. Constructing a Weighted Image.* In this section, the weighted image is obtained by unifying calculation of the Itti visual saliency map and the contourlet transform decomposition coefficients, which is shown Algorithm 1. The specific calculation steps are as follows.

*(1) Calculate the Itti Visual Saliency Map M.* The Itti model [29] mainly utilizes techniques such as binary Gaussian pyramid, center-surround computation, cross-scale computation, and linear fusion to extract visual saliency map. Generally, the Itti model calculation is divided into four steps. The first step is to extract the saliency map, denoted as $M_{\text{colors}}$, through techniques such as Gaussian pyramid, center-surrounding computation, and cross-scale computation. The second step is to extract the intensity saliency map, $M_{\text{intensity}}$, which is similar to the calculation process of color saliency map extraction. Likewise, the third step is to extract the saliency map of the orientation, $M_{\text{orientation}}$. Finally, the above three saliency maps are combined to

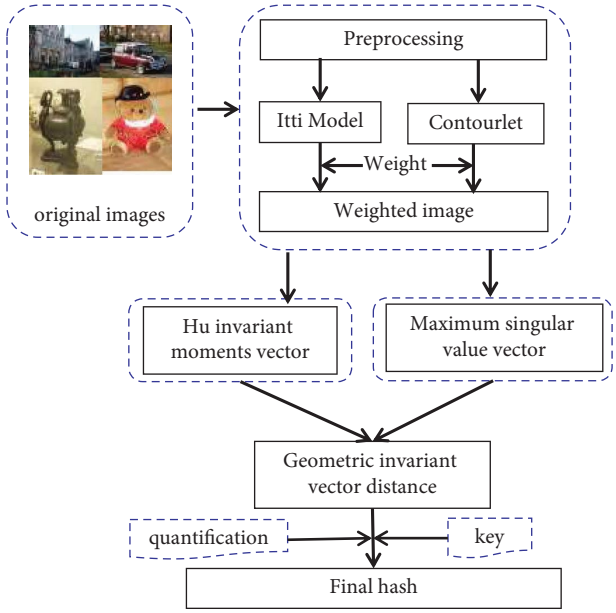



Figure 2: The framework of hashing generation.

**Input**: the original image $I_0(x, y)$
**Output**: the weighted image
**Begin**
  **begin preprocessing**
    standardize the image sized $n \times n$
    Gaussian low-pass filter
    return $I_1(x, y)$
  **end preprocessing**
  $M$ = Itti_Model ($I_1(x, y)$);
  **begin calculating N**
    $I_2$ = HSI_Mode ($I_1(x, y)$)
    $N$ = Contourlet_transform ($I_2$)
  **end calculating N**
  standardize M and N, return J and K
  Return the weighted image $I = J \times K$
**End**

Algorithm 1: Weighted image calculation.

generate the final visual saliency map $M$. The calculation formula is shown as

$$M = \frac{\left(M_{\text{colours}} + M_{\text{intensity}} + M_{\text{orientations}}\right)}{3}. \quad (3)$$

*(2) Calculate the Contourlet Transform Coefficient N.* In this study, the contourlet transform [30] is selected for weighting because of its good suitability of sparse representation of 2D images. On the one hand, contourlet transform has the multiresolution time-frequency analysis features of wavelet transform. On the other hand, the contourlet transform also has good anisotropy, which can approximate singular curves with the least coefficients along the edge of the image contour. Due to the better sparse representation of the image, rudimentary data compression can be achieved, and it is less affected by common digital operations, for instance, noise pollution, low-pass filtering, and JPEG compression. According to formula (4), the preprocessing image $I_1(x, y)$ is calculated to obtain $I_2$, which is used as the input of the contourlet transformation. Let $N$ be the contourlet transform coefficient:

$$I_2 = \frac{(R + G + B)}{3}. \quad (4)$$

*(3) Calculate the Weighted Image.* Assuming that the size of the visual saliency map $M$ is $S \times S$ and the contourlet coefficient $N$ is $P \times Q$, in order for calculating the weighted

image, the size of $M$ and $N$ is sampled as $n \times n$ by the bicubic interpolation. Let $J$ and $K$ be the sampled versions of $M$ and $N$; therefore, the weighted image I can be calculated by

$$I(x, y) = J(x, y) \times K(x, y), \tag{5}$$

where $I(x, y)$, $J(x, y)$, and $K(x, y)$ represent the elements of the matrices $I$, $J$, and $K$ at row $x$ and column $y$, respectively.

The weighted image is randomly divided into blocks, and the random key is denoted as $K_1$. Let $B_k, k = 1, 2, \cdots, L$ be the $k$th block and the size of $B_k$ is $b \times b$.

*3.1.3. Hu Invariant Moment Vector Extraction.* For each block $B_k$, seven Hu invariant moments are calculated, and we get the column feature vector $\phi_k = [\varphi_1(k), \varphi_2(k), \cdots, \varphi_7(k)]^T$. Let $\Phi = [\phi_1, \phi_2, \cdots, \phi_L]_{7\times L}$ be the invariant moment feature matrix of the entire image. To build a normalized matrix for further calculation, data normalization is carried out. Let $\Phi_j = [\Phi_j(1), \Phi_j(2), \cdots, \Phi_j(L)]$ be the $j$th $(1 \le j \le t)$ row of $\Phi$. Each element $\Phi_j(i)$ is normalized in the $j$th row $\Phi_j$ as

$$\Phi_j(i) = \frac{\Phi_j(i) - \mu_j}{\delta_j}, i = 1, 2, \cdots, L. \tag{6}$$

In formula (6), $\mu_j$ and $\delta_j$ are the mean and the standard deviation of all elements of $\Phi_j$.

Finally, the normalized Hu invariant moment vector is obtained, which is denoted as $\Psi = [\psi_1, \psi_2, \cdots, \psi_L]_{7\times L}$.

*3.1.4. The Largest Singular Value Vector Extraction.* Similarly, for each block $B_k$, SVD is performed. We extract the singular vector corresponding to the maximum eigenvalue, denoted as $U_1(k)(1 \le k \le L)$. Let $U = [U_1(1), U_1(2), \cdots, U_1(L)]$ be the largest singular value vector. In the same way, the vector is normalized as formula (6), and the normalized largest singular value vector is obtained, which is denoted as $V = [V_1, V_2, \cdots, V_L]_{b\times L}$.

*3.1.5. Calculating Geometric Invariant Vector Distance.* Through the above steps, we get the feature vectors $\Psi$ and $V$. Since the vector distance is invariant to the digital operations on the content of the image, we use the invariant vector distance to construct perceptual hashing, which is extracted in detail as follows.

(1) Calculating the geometric invariant vector distance of the matrix $\Psi$;

For the matrix $\Psi_{7\times L}$, in order to calculate the geometric invariant vector distance, we provide a reference vector $\Psi_0 = [\Psi_0(1), \Psi_0(1), \cdots, \Psi_0(7)]^T$, where $\Psi_0(j)(1 \le j \le 7)$ is the mean of the $j$th row.The Euclidean distance between the $k$th column vector $\psi_k = [\psi_1(k), \psi_2(k), \cdots, \psi_7(k)]^T$ and the $\Psi_0$ column is defined as

$$d_1(k) = \sqrt{\sum_{j=1}^{7} \left(\psi_j(k) - \Psi_0(j)\right)^2}. \tag{7}$$

After calculation, the geometric invariant vector distance of Hu invariant moment eigenvector is obtained, which is denoted as $D_1 = [d_1(1), d_1(2), \cdots, d_1(L)]$.

(2) Calculating the geometric invariant vector distance of the matrix $V$:

Also, for the matrix $V = [V_1, V_2, \cdots, V_L]_{b\times L}$, to calculate the geometric invariant vector distance, we provide a reference vector $V_0 = [V_0(1), V_0(1), \cdots, V_0(n)]^T$, where $V_0(j)(1 \le j \le b)$ is the mean of the jth row. The Euclidean distance between the $k$th column vector $V_k = [V_1(k), V_2(k), \cdots, V_b(k)]^T$ and the $V_0$ column is defined as

$$d_2(k) = \sqrt{\sum_{j=1}^{b} \left(\psi_j(k) - V_0(j)\right)^2}. \tag{8}$$

After calculation, the geometric invariant vector distance of the maximum singular vector is obtained, which is denoted as $D_2 = [d_2(1), d_2(2), \cdots, d_2(L)]$.

After calculating through step (1) and (2), we get the feature vector $D = [D_1, D_2]_{2L}$.

(3) Encryption:

In order to improve the security of the algorithm, we use the chaotic encryption algorithm to encrypt the vector $D$. Assuming that $K_2 \in (0, 1)$ is a secret key shared by the sender and the receiver, the logistic equation is shown as

$$k_{n+1} = \mu k_n(1 - k_n), 3.5699 < \mu < 4. \tag{9}$$

Let $k_1 = K_2, k_2 = \mu k_1(1 - k_1), \cdots, k_{n+1} = \mu k_n(1 - k_n)$, and then the key vector Key $= [k_1, k_2, \cdots, k_{2L}]$ is obtained.

Let $Y = D \times \text{Key} = [y_1, y_2, \cdots, y_{2L}]_{2L}$, where $Y$ is the hashing sequence after encryption and randomization.

(4) Quantization:

Since the encrypted eigenvalues are all floating-point data, in order to reduce the storage space and improve the efficiency of image authentication, the eigenvector $Y$ is quantized by

$$h(i) = [y_i * 10 + 0.5], 1 \le i \le 2L, \tag{10}$$

where $[\cdot]$ is the round-up or round-down operation.

Finally, the final hashing sequence is generated, denoted as hash $= [h_1, h_2, \cdots, h_{2L}]_{2L}$. In order to determine the hashing length in the binary form, the hashing values of 1000 different images are calculated and analyzed to determine the number of bits occupied by each hashing value, which is discussed in Section 4.6.

From Section 3.1.3 to Section 3.1.5, we can calculate the final hash sequence by Algorithm 2.

*3.2. Image Copy Detection.* Let IS be the image set, hash $(\cdot)$ be the hash function, and $\varphi(\bullet)$ be the content-preserving operation function. There are two different images $\forall x, y \in IS$ with different perception contents, and $h_x = \text{hash}(x)$ and $h_y = \text{hash}(y)$ represent the hash sequence of the image $x$ and $y$, respectively. Let $x' = \varphi(x)$ and $h_{x'} = \text{hash}(x')$ represent the hash of the image $x'$.

Euclidean distance is used to measure the difference between two image hashes during image copy detection. When the Euclidean distance between the two different images is greater than the threshold $T$, that is, $D(h_x, h_y) > T$, it is determined that the two images are different perceptual contents. When the Euclidean distance between the origin image and its duplicate images is less than the threshold $T$, that is, $D(h_x, h_{x'}) < T$, it is determined that these duplicate images are from a copyright image.

## 4. Experiment Results

The proposed algorithm is programmed in Matlab language and implemented on the platform of Windows 2010 and MATLAB 2018b, in which the CPU is i7-4770k @ 3.50 GHz and the memory is 8.00 GB. The image datasets used in the experiment includes 1022 images. There are 22 images sized $768 * 512$ or $512 * 768$ from Kodak datasets [31], and the other 1000 images sized $512 * 384$ or $384 * 512$ are from the UCID [32]. The simulation results and analysis are as follows.

*4.1. Experimental Parameters.* The experimental parameters are as follows. The original image is interpolated into a fixed size $256 \times 256$, that is, $n = 256$, and the size of the random block size is $60 \times 60$, that is, $b = 60$. The number of blocks is 50, that is, $L = 50$. The size of Gaussian low-pass filter convolution mask is $3 \times 3$, and the standard deviation is 0.5. The random block key is 100, and the initial key of the chaotic encryption of the eigenvector is 0.27, that is to say, the key pair $(K_1, K_2)$ is (100, 0.27). According to the process of the previous Algorithm 2, the length of the hashing sequences is equal to twice the number of blocks, that is, 100 integers. Next, the simulation results and analysis of the experiment are given below. Section 4.2 verifies robustness, Section 4.3 discusses discrimination, Section 4.4 analyzes the hashing security, Section 4.5 explores the influence of different parameters, Section 4.6 probes hash length and storage cost, Section 4.7 analyzes which component is crucial for the proposed algorithm, and Section 4.8 discusses the performance comparison of other state-of-the-art algorithms.

*4.2. Robustness.* To evaluate the performance of the raised algorithm, 22 images sized $768 \times 512$ or $512 \times 768$ in the Kodak dataset were subjected to robust attacks using Photoshop, Matlab, and StirMark [33] to acquire their duplicate images. These robust attacks are composed of a wide range of operations with perceptual content unaltered, and the details are shown in Table 1, containing the using tools, operation types, the parameters and the values of parameters, and the number of each operation. Therefore, the total number of robustness attacks utilized in the experiment is 69. In other words, there are 69 copy versions of each original image after the robustness attacks. Consequently, there are $22 * 69 = 1518$ duplicate images, bringing the total number of images to $1518 + 22 = 1540$. We extract hashing sequences of the 22 test images and their copy images, and calculate the Euclidean distance between each original image and its duplicate versions. In this experiment, we select the threshold $T = 70$ (for the selection of threshold, please see the relevant instructions in Section 4.3). As described in Section 3.2, when the Euclidean distance between the origin image and its duplicate images is less than the threshold $T$, that is, $D(h_x, h_{x'}) < T$, it is determined that the duplicate images are from a copyright image.

The results of the Euclidean distance for each operation are listed in Table 2, including the maximum value, the minimum value, the mean value, and the standard deviation. The variation curves of Euclidean distance mean under different robustness operations and parameters are revealed in Figure 3, where the abscissa represents the different parameter settings under each digital processing operation, and the ordinate represents the Euclidean distance mean. It is clear from Table 2 and Figure 3 that the algorithm has good robustness to the majority of robust operations, especially for JPEG compression, affine transformation, salt pepper noise, speckle noise, gamma correction, watermark embedding, scaling, contrast and brightness changes, and the maximum values of European distances corresponding to these operations are lower than the threshold ($T = 70$), and their average values are lower than 21. In addition, it can be seen from Table 2 and Figure 3(g) that although the maximum value of Gaussian noise is 75.1961, the average value range is [20.1225, 37.5264], and the overall average value is 29.2106. In this experiment, only three image pairs have Euclidean distances exceeding 70, therefore, the proposed algorithm also shows good performance against Gaussian noise. However, for rotation transformation, it has poor adaptability. Table 3 lists the statistical analysis results corresponding to different rotation angle ranges of 22 images, including the number of distances exceeding the threshold ($T \geq 70$), mean value, and $P_{\text{TPR}}$ (shown in formula (11)). It can be seen from Tables 2, 3 and Figure 3(j) that when the rotation angle range is less than 2 degrees, the proposed algorithm has strong adaptability, and the $P_{\text{TPR}}$ can reach 96.59%, with merely six pairs exceeding 70. However, when the rotation angle is greater than 3 degrees, the performance of the algorithm decreases, and the $P_{\text{TPR}}$ is less than 90%.

Therefore, the experiments show that the raised algorithm has good robustness to the majority of robust operations.

**Input**: the weighted image I
**Output**: the hashing sequence
**Begin**
  **for** 1: k (maximum number of blocks)
    **for** 1: b (block size)
      $B_k$ = overlappingblock (I) with secret key $K_1$;
    **end**
  **End**
  **While** $B_k$
$\phi_k$ = Hu_Feature($B_k$);
  **End**
  Hu_Feature_Vector $\Psi_{7\times L}$ =Normalize ($\phi$);
  **While** $B_k$
    $U(k)$ = SVD_Feature ($B_k$);
  **End**
  Hu_Feature_Vector $V_{b\times L}$ = Normalize($U$);
  D1 = geometric_invariant_distance ($\Psi$);
  D2 = geometric_invariant_distance ($V$);
  $D = [D_1, D_2]_{2L}$;
  Y = chaotic_encryption (D) with secret key $K_2$;
  Hash = quantize (Y);
**End**

Algorithm 2: The hashing extracting method.

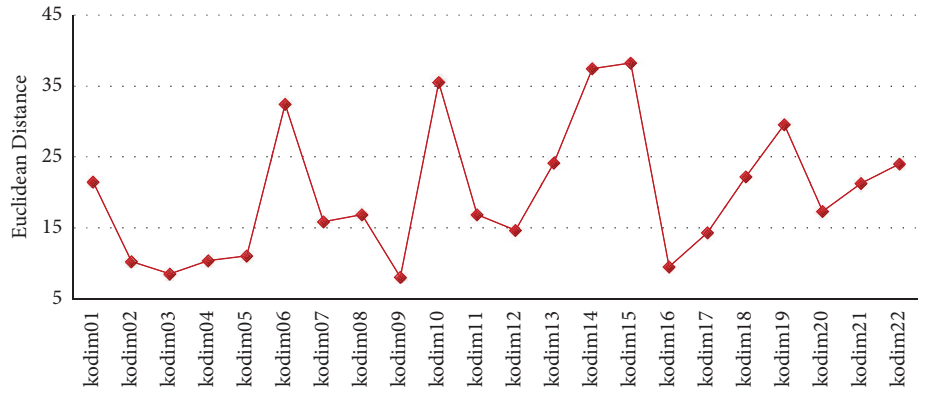
Table 1: The operation types and parameters adopted in the experiment.

| Tool | Operation types | Parameters | Parameters values | Number |
|---|---|---|---|---|
| Photoshop | Contrast adjustment | Contrast scale | ±10, ±20, ±30 | 6 |
| Photoshop | Brightness adjustment | Brightness scale | ±10, ±20, ±30 | 6 |
| Matlab | Speckle noise | Variance | 0.002, 0.004, 0.006, 0.008, 0.01 | 5 |
| Matlab | Salt and pepper noise | Density | 0.002, 0.004, 0.006, 0.008, 0.01 | 5 |
| Matlab | Gaussian noise | Variance | 0.002, 0.004, 0.006, 0.008, 0.01 | 5 |
| Matlab | Gamma correction | $\gamma$ | 0.5, 0.7, 0.9, 1.1, 1.3 | 5 |
| StirMark | JEPG | Quality factor | 25, 30, 35, 40,. . ., 80, 90, 100 | 10 |
| StirMark | Watermark embedding | Strength | 10, 20, 30,. . ., 90, 100 | 10 |
| StirMark | Image scaling | Ratio | 0.5, 0.75, 0.9, 1.1, 1.5, 2.0 | 6 |
| StirMark | Image rotation | Angle | ±1, ±2, ±3, ±4, ±5 | 10 |
| StirMark | Affine transformations | Transformation matrix | [1 0 0 0.01 1 0] | 1 |
| | | Total | | 69 |

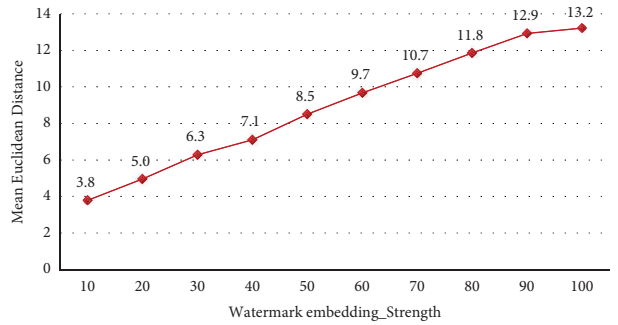
Table 2: Statistical results of the Euclidean distance under different operations.

| Operation | Maximum value | Minimum value | Mean value | Standard deviation |
|---|---|---|---|---|
| Contrast adjustment | 27.641 | 3.317 | 8.947 | 4.564 |
| Brightness adjustments | 33.690 | 2.646 | 10.583 | 6.396 |
| Speckle noise | 42.954 | 4.123 | 13.747 | 9.781 |
| Salt and pepper noise | 48.539 | 3.873 | 13.350 | 9.321 |
| Gaussian noise | 75.196 | 7.550 | 29.211 | 16.334 |
| Gamma correction | 58.181 | 4.900 | 17.492 | 12.033 |
| JEPG | 67.985 | 4.123 | 17.099 | 10.337 |
| Watermark embedding | 34.627 | 1.414 | 8.906 | 6.382 |
| Image scaling | 29.069 | 3.000 | 10.985 | 5.729 |
| Image rotation | 111.557 | 13.038 | 53.609 | 21.428 |
| Affine transformations | 38.223 | 8.062 | 20.012 | 9.566 |

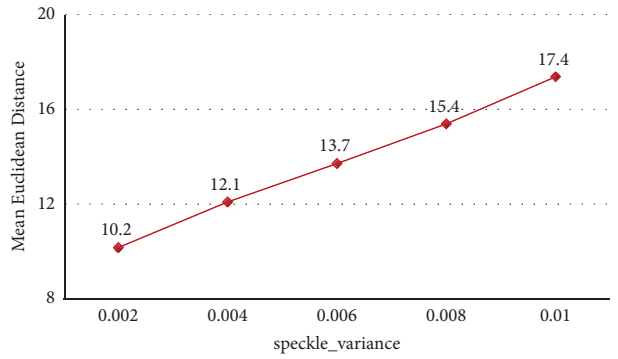
*4.3. Discrimination.* In order for verifying the discrimination, 1000 images which are visually different are selected from UCID image databases. To begin with, we attain the perceptual hashing sequences of these images using the raised algorithm, and we match pairwise among all the different images at random and calculate the Euclidean



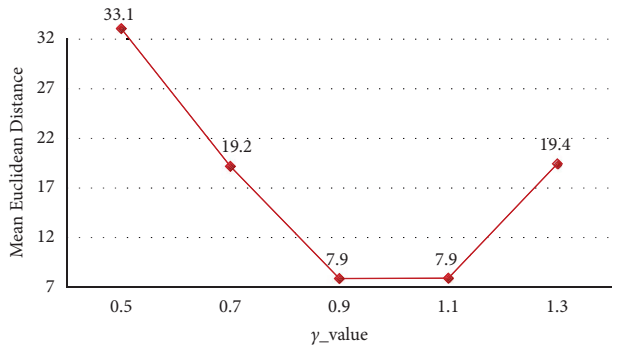

(a)




(b)

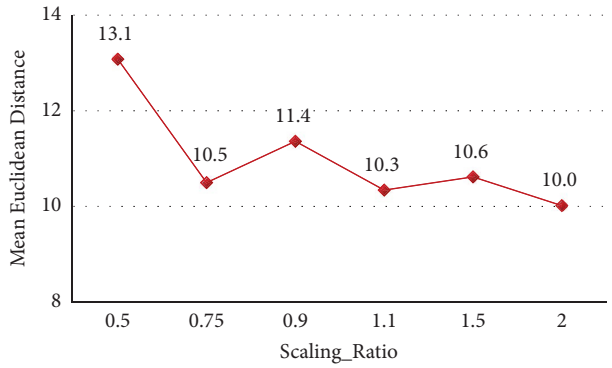



(c)

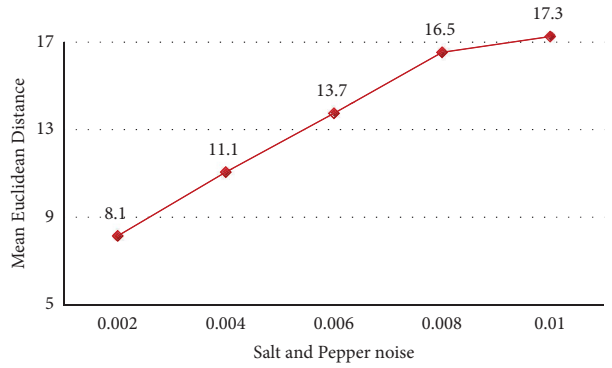



(d)

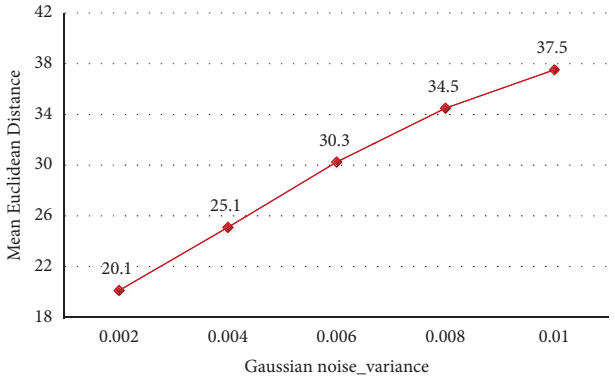
Figure 3: Continued.



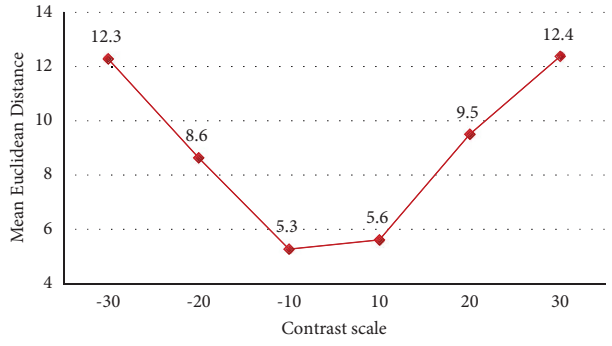

(e)




(f)




(g)




(h)

Figure 3: Continued.

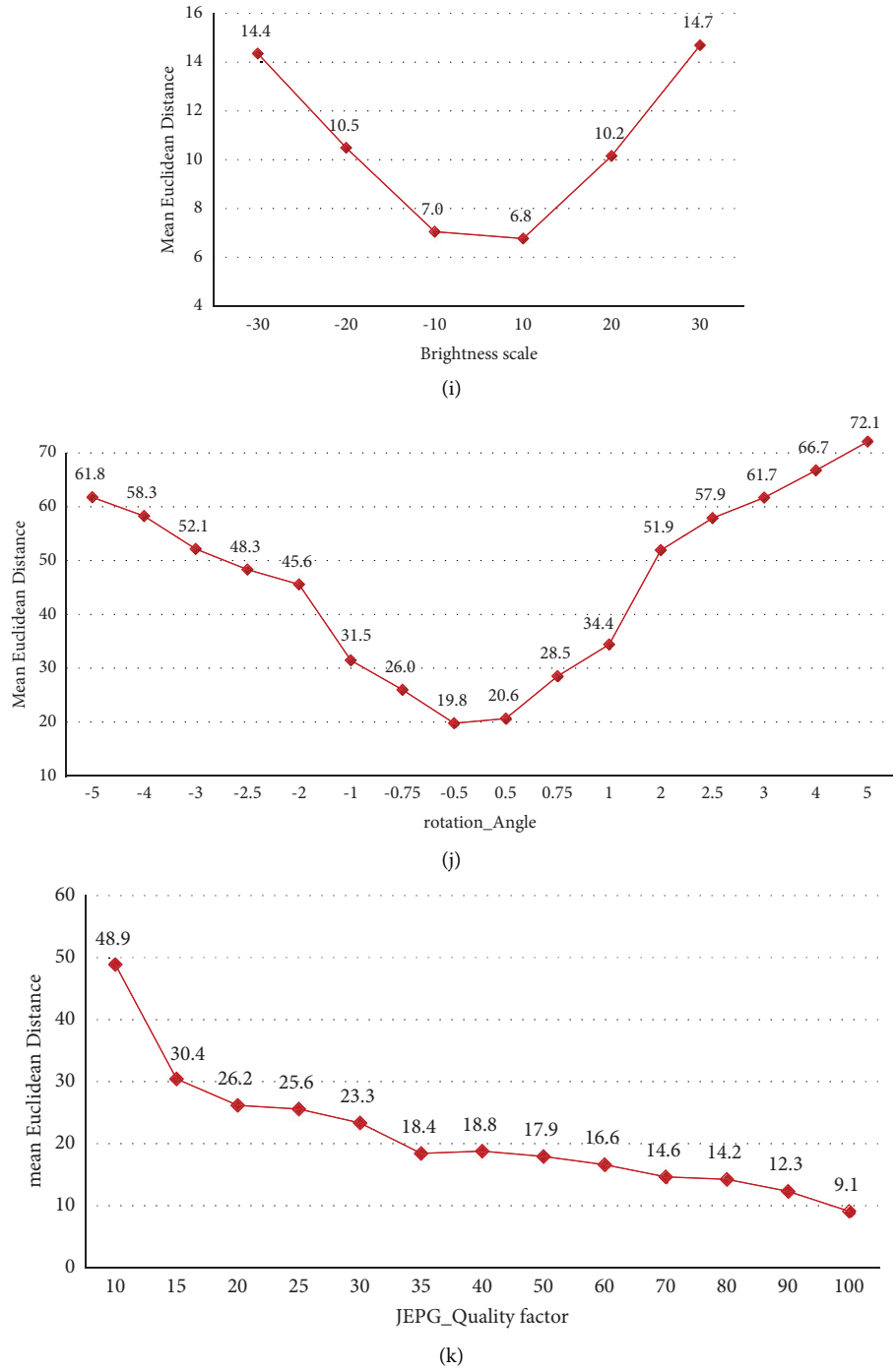



Figure 3: The mean value of the normalized hamming distance under different operations: (a) affine transformation, (b) watermark embedding, (c) speckle noise, (d) gamma correction, (e) scaling, (f) salt and pepper noise, (g) Gaussian noise, (h) contrast adjustment, (i) brightness adjustment, (j) rotation, and (k) JEPG compression.

distance of each pair. After that, we get $1000 \times (1000 - 1)/2 = 499500$ sets of Euclidean distances. In the end, the frequency statistic of these groups of distances is counted, and we plot the frequency histogram which reveals the total intensity distribution of these Euclidean distances, as shown in Figure 4. It is apparent that the result of the experiment nearly is in line with the normal distribution, where the mean value is 120.7119, standard deviation is 29.1825 , the maximum value is 296.2836, and the minimum value is 31.8434.

In order to determine a reasonable threshold, the true positive rate($P_{\mathrm{TPR}}$), the false positive rate($P_{\mathrm{FPR}}$), and the total error rate ($1 - P_{\mathrm{TPR}} + P_{\mathrm{FPR}}$) under a variety of thresholds are calculated and itemized in Table 4, with detailed definitions of $P_{\mathrm{TPR}}$ and $P_{\mathrm{FPR}}$ shown in formulas (11) and (12). The range of the threshold $T$ is between 50 and 150, with increasing step

Table 3: Statistical results of rotation operations.

| Rotation angle range | [−5,+5] | [−4,+4] | [−3,+3] | [−2,+2] | [−1,+1] |
|---|---|---|---|---|---|
| Number ($T \geq 70$) | 58 | 41 | 27 | 6 | 0 |
| Mean value | 53.6090 | 43.1023 | 39.8683 | 32.2923 | 26.8037 |
| $P_{TPR}$ | 0.8352 | 0.8669 | 0.8977 | 0.9659 | 1.0000 |

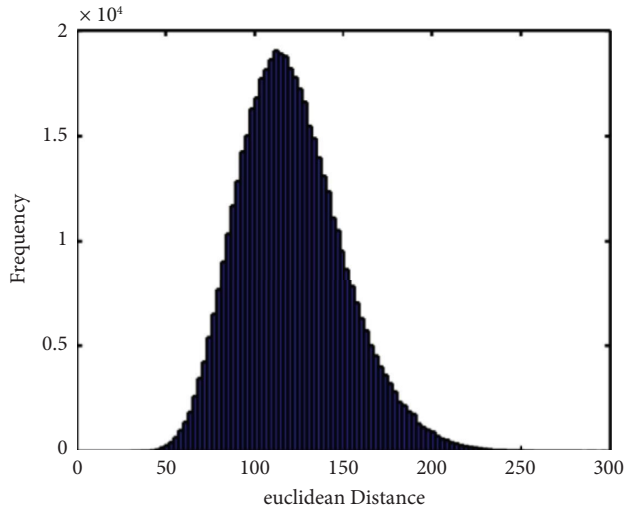



Figure 4: Histogram of the Euclidean distance of different images.

Table 4: The true positive rate, the false positive rate, and the total error rate under different thresholds.

| Threshold | $P_{TPR}$ | $P_{FPR}$ | $1 - P_{TPR} + P_{FPR}$ |
|---|---|---|---|
| 100 | 0.99539 | 0.25120 | 0.25265 |
| 95 | 0.99341 | 0.19220 | 0.14392 |
| 90 | 0.98880 | 0.14050 | 0.15173 |
| 85 | 0.98353 | 0.09760 | 0.11405 |
| 80 | 0.97892 | 0.06410 | 0.08521 |
| 75 | 0.97563 | 0.03930 | 0.06365 |
| 70 | 0.96574 | 0.02240 | 0.05664 |
| 65 | 0.95389 | 0.01150 | 0.05761 |
| 60 | 0.94466 | 0.00550 | 0.06079 |
| 55 | 0.93544 | 0.00220 | 0.06676 |
| 50 | 0.91304 | 0.00076 | 0.08772 |

size of 5. The threshold criterion that we selected here is to minimize the total error rate. It is observed from Table 4 that when the threshold $T = 70$, the $P_{TPR}$ stands at 0.96574, while $P_{FPR}$ is merely 0.0224, with the total error rate reaching the minimum value (0.0566). On the other hand, when the threshold is lower than or higher than 70, the total error rate gradually increases and converges to the minimum value only when $T = 70$. Remarkably, the overall performance of the algorithm can be evaluated by the total false rate. Therefore, this algorithm can choose $T = 70$ as a recommended threshold.

Obviously, a small threshold means a low collision probability, that is, good discrimination. A large threshold is helpful for improving the robustness of the algorithm, but it will inevitably reduce the discrimination performance. Therefore, in practice, the threshold can be selected on the basis of the specific application scenarios to obtain the balance between discrimination and robustness. For image similar detection, for example, the threshold $T$ can be taken at 50. Under the circumstance, the $P_{FPR}$ reaches the order of $10^{-4}$ magnitude, that is to say, when there are $10^4$ images received, one pair of diverse images regarded as visually identical images would probably occur. Therefore, it is believed that this method can guarantee the discrimination of different perceptual images.

*4.4. Security Experiment.* In this paper, the key K1 is used for random selection of image blocks, and the initial secret key K2 is used for chaotic encryption, that is, the key pair (K1, K2) is taken advantage of enhancing the safety of the algorithm. The key pair is different and the resulting image hash should also be different. For the sake of verifying the dependence of the algorithm on the key pair, we firstly make

use of the initial key pair (201, 0.27) to generate the hash of an image, and secondly, select 200 different key pairs (K1, K2) at random to extract the hashes of the same image. Next, the Euclidean distance of 201 hash values is matched, and a total of 19,900 matching results are generated. The result of experiment is given in Figure 5. Clearly, the least Euclidean distance is 155.5796, with being far larger than the given threshold. It illustrates that hash sequences generated by various key pairs of the same image will not have false matches. Therefore, when the key pair is unacquainted, it is impossible to forge the correct image hashing, which heightens the safety of the algorithm.

*4.5. Influence of Different Parameters on Hashing Performance.* This section mainly explores the influence of the image block size (denoted as $b$) and the number of random blocks (denoted as $L$) on the classification performance of the algorithm with respect to robustness and discrimination. The image databases of Sections 4.2 and 4.3 are still employed in this experiment. In the experiments, only the parameter discussed currently is changed, and the other parameters maintain unchangeable. In this section, the famous receiver operating characteristic (ROC) [34] curve is exploited to analyze the classification performance of the algorithm. In the ROC curve, the $y$-axis coordinate usually represents the true positive rate $P_{\mathrm{TPR}}$, while the $x$-axis coordinate represents the false positive rate $P_{\mathrm{FPR}}$. The specific calculation formulas of $P_{\mathrm{TPR}}$ and $P_{\mathrm{FPR}}$ are defined as follows:

$$P_{\mathrm{TPR}}\left(D_{\mathrm{hash}} \le T_{\mathrm{same}}\right) = \frac{n_1}{N_1}, \tag{11}$$

$$\mathrm{P}_{\mathrm{FPR}}\left(\mathrm{D}_{\mathrm{hash}} \le \mathrm{T}_{\mathrm{diff}}\right) = \frac{n_2}{N_2}, \tag{12}$$

where $n_1$ and $N_1$, respectively, represent the number of the pairs of copy images estimated as visually identical images and the whole pairs of visually identical images, $n_2$ and $N_2$, respectively, represent the number of the pairs of diverse images regarded as visually identical images and all the pairs of visually different images. Apparently, $P_{\mathrm{TPR}}$ and $P_{\mathrm{FPR}}$, respectively, represent the robustness and discrimination of the algorithm. Therefore, the ROC curve is closer to the upper-left corner (i.e., the smaller value of $P_{\mathrm{FPR}}$, the larger value of $P_{\mathrm{TPR}}$), then the classification performance of the algorithm is better. For quantitative comparison, ROC curve area (AUC) [34] can be regarded as a judgment standard. The range of AUC value is (0, 1), and a larger AUC stands for better classification performance. The influence of parameter $L$ and $b$ on the algorithm is discussed.

*4.5.1. The Number of Random Blocks.* Figure 6 is the contrast chart of ROC curves under diverse $L$ values when $b = 60$, including $L = 30$, 40, 50, 60, and 70. It is obvious that all ROC curves get close to the upper-left corner, indicating that these $L$ values can make the presented algorithm commendably balanced as for robustness and discrimination. The ROC curves in the left-top corner are blown up, as displayed in the small graph below the middle. Obviously, the ROC curve

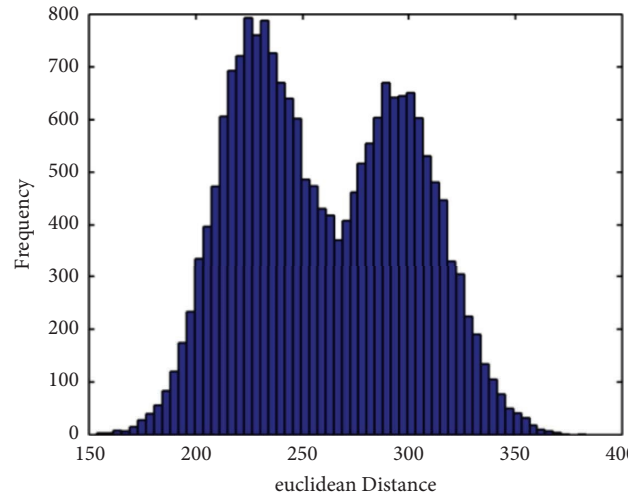



Figure 5: Histogram of the Euclidean distance of different key pairs on the same image.

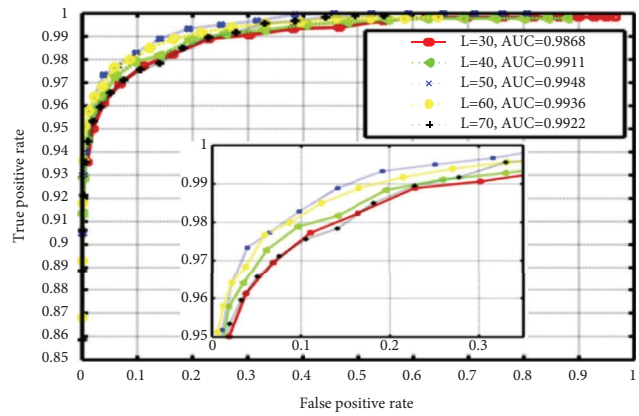



Figure 6: Comparison of ROC curves among different $L$ values.

with $L = 50$ is closest to the upper-left corner, and the corresponding AUC value is 0.9948, which is higher than the AUC value of the ROC curve with other $L$ values. Therefore, the number of random blocks $L$ is selected as 50. It can be seen from Figure 6 that when $L = 30$ and $L = 70$, the ROC curves appear alternately, and the corresponding AUC values are 0.9868 and 0.9922, respectively, which are also lower. This is because when the block size $b$ is fixed, a small number of blocks will cause the key features of the image to not be collected, and the image cannot be expressed well, resulting in performance degradation. If the value of $L$ is too large, some unnecessary redundant features will be introduced, leading to overfitting. In addition, the ROC curves of $L = 30$, 40, 50, 60, and 70 are very close, and the corresponding AUC values are also very close, which shows that the number of random blocks $L$ has little influence on the algorithm, and it also reflects the good stability of the algorithm on the other hand.

*4.5.2. The Image Block Size.* Figure 7 is a comparison graph of ROC curves under diverse $b$ values when $L = 50$, including $b = 30$, 40, 50, 60, and 70. It is distinct that all ROC curves

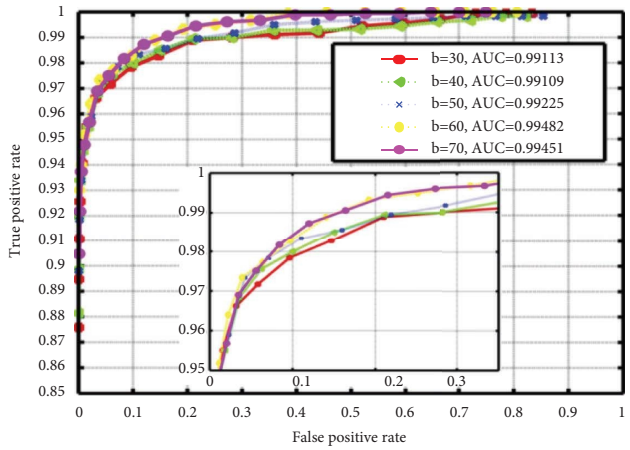



Figure 7: Comparison of ROC curves among different $b$ values.

approach the upper-left corner, manifesting that these $b$ values make the raised algorithm magnificently balanceable in regard to robustness and discrimination. The ROC curves in the upper-left corner are blown up, as exhibited in the small figure below the middle. Obviously, the ROC curves of $b = 60$ and $b = 70$ appear alternately and are significantly higher than the ROC curves of $b = 30$, 40, and 50. This is because when the number of random blocks $L$ is fixed, the smaller the block is, the easier it is to cause the loss of image features, which cannot express the image well, resulting in a decrease in classification performance. If the value of $b$ is too large, some repeated image regions will be introduced, leading to feature redundancy and overfitting. From Figure 7, the AUC value of $b = 60$ is 0.99482 higher than that of $b = 60$, whose AUC value is 0.99451. Therefore, we choose the image block size $b = 60$.

*4.6. The Length of Hashing.* The storage cost of the hashing is analyzed though making use of the hashing sequences extracted in discriminative experiments in Section 4.3. As just mentioned before, every hashing sequence contains 100 elements, so the total number of the hashing elements is $1000 \times 100 = 100000$. The hashing value distribution curve of each image is shown in Figure 8. According to the calculation and statistics, the least value is 1, the biggest value is 183, and the average value is 28.3104. We calculate the frequency distribution of these hashing element values and get a frequency distribution bar graph as shown in Figure 9, where the $x$-axis is the value range of the hash element and the $y$-axis is its frequency. From Figures 8 and 9, it is clear that the hashing element values are distributed between [0, 128), accounting for 99.985%, and only 15 hash values are between 128 and 183, that is, (128, 183). Since 8 bits can represent unsigned integer in the interval (0.255), each hashing element of the algorithm only needs 8 bits to store. It can be seen that the storage cost of the hashing of the proposed algorithm is 800 bits.

*4.7. Component Analysis.* The algorithm in this paper mainly includes two steps, one is weighted image construction (see Algorithm 1 for details), and the other is hashing feature

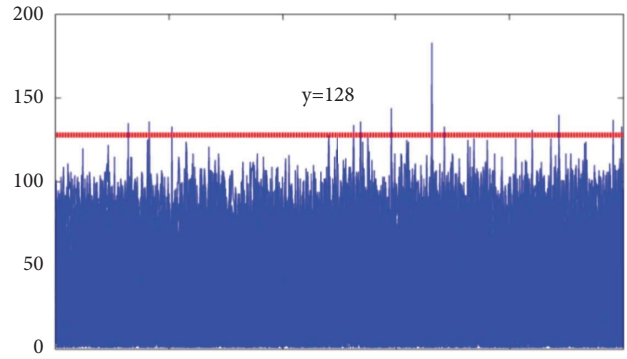



Figure 8: The distribution of hashing value.

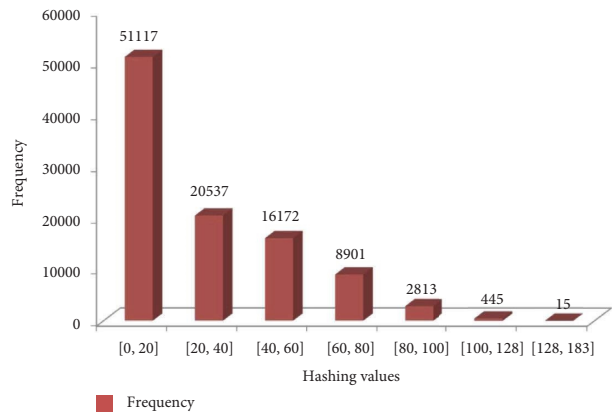



Figure 9: The bar chart of frequency distribution of hash value.

extraction (see Algorithm 2 for details). To begin with, for Algorithm 1, we test the impact of each single component (the Itti visual model or contourlet transform (CT)) on the algorithm performance, respectively, with the remaining Algorithm 2 unchanged. Similarly, for algorithm 2, we mainly check out the influence of each single component (the geometric invariant vector distance of Hu eigenvector or SVD eigenvector) respectively, keeping Algorithm 1 fixed. Because different components directly affect the Euclidean distance of the image hashing, different threshold values are selected in the experimental tests of different components, and the selection standard is also to minimize the total error rate. Table 5 shows $P_{\mathrm{TPR}}$, $P_{\mathrm{FPR}}$, the total error rate, and corresponding thresholds under different components. It is clear that the overall performance of SVD component is the best among the four components, with $P_{\mathrm{TPR}}$ reaching 0.9447, $P_{\mathrm{FPR}}$ standing at 0.0090, and a total error rate of 0.0643. The Itti component is second only to SVD. However, the overall performance of the two factors, contourlet and Hu component, is poor. Figure 10 shows the ROC curves of four components, including the Itti visual model, contourlet transform, Hu feature vector, and SVD feature vector. It can also be seen from ROC curves that SVD feature vector and the Itti visual model are two important components of the proposed algorithm, and their corresponding ROC curves are very close (AUC values are 0.9938 and 0.9904, respectively), which are significantly higher than the ROC curves of

Table 5: Correct detection rate and collision rate under different factors.

| Components | $P_{FPR}$ | $P_{TPR}$ | $1-P_{TPR}+P_{FPR}$ | Threshold |
|---|---|---|---|---|
| Itti | 0.0235 | 0.9526 | 0.0709 | 105 |
| CT | 0.0102 | 0.8933 | 0.1169 | 11 |
| SVD | 0.0090 | 0.9447 | 0.0643 | 55 |
| Hu | 0.0276 | 0.9144 | 0.1132 | 19 |
| All | 0.0224 | 0.9657 | 0.0566 | 70 |

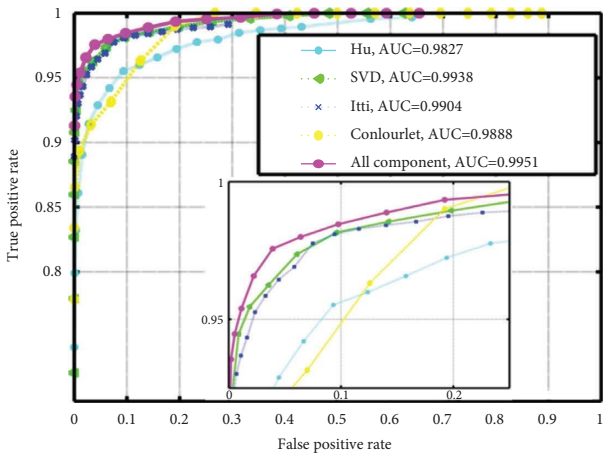



Figure 10: Comparison of ROC curves under different factors.

the other two factors, contourlet transform, and Hu eigenvector. But no matter Table 5 or Figure 10, we can clearly find that the performance of the proposed algorithm is the best when the four components are combined. In the experiment, it is also found that the Euclidean distances of hashing sequence extracted by contourlet transform and Hu feature vector have little gap in robustness and discrimination. To sum up, SVD feature vector and the Itti visual model are two key components of the proposed algorithm.

*4.8. Performance Comparisons of Different Algorithms.* For the sake of illustrating the superiority of this algorithm, it is compared with some state-of-the-art algorithms which were recently issued in academic journals or conference. The comparison algorithms contain geometric invariant vector distance (GIVD) [21], visual attention model and invariant moments (VAM and IM) [22] and hybrid features hashing [35]. The experimental image library is the same as Sections 4.2 and 4.3. For impartial comparison, all parameter settings are in accordance with their original settings. In this algorithm, $L = 50$, $b = 60$, and the key pair (100, 0.27) are selected for the comparison among algorithms.

Figure 11 shows the ROC curves of different algorithms. By comparison, it can be seen clearly that the AUC value of the proposed algorithm is 0.9951, and the AUC values of the GIVD algorithm, the algorithm based on VAM and IM and the hybrid features algorithm are 0.9887, 0.9914, and 0.9946, respectively. Obviously, the AUC value of the proposed algorithm is a bit higher than that of the comparison algorithms. Therefore, by comparing the ROC curves and

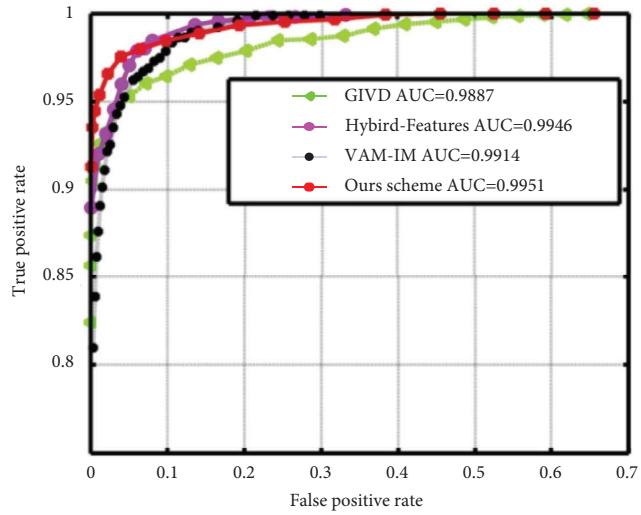



Figure 11: Comparison of ROC curves among different algorithms.

Table 6: Performance comparisons among different algorithms.

| Algorithm performance | Time (s) | AUC | Hash length (bit) |
|---|---|---|---|
| Ours scheme | 0.1710 | 0.9951 | 800 |
| GIVD [21] | 0.0594 | 0.9887 | 700 |
| Hybrid features [35] | 29.9680 | 0.9946 | 3328 |
| VIM_IM [22] | 0.1610 | 0.9914 | 1152 |

AUC values, we find out that the proposed algorithm outperforms other algorithms in regard to classification performance.

Otherwise, we also analyze other properties of different algorithms, as displayed in Table 6, such as running time, the AUC value, and the length of the hashing. For calculation of running time, in the first place, we recorded the total time spent on generating hash sequence from 1000 different images and then calculated the average time cost on running an image. It can be seen from Table 6 that the proposed algorithm is faster than the hybrid features algorithm and slower than the GIVD algorithm and is equal to the VIM_IM algorithm. From the overall perspective of running time, the length of the hash, and classification performance, the proposed algorithm has better combination property.

## 5. Conclusion

In this study, an image hashing algorithm is proposed based on the Itti visual saliency model and geometric invariant vector distance, combining the features of the frequency domain and the spatial domain. The Itti visual saliency model and low-frequency subband of the contourlet transform are put into use in constructing a weighted image, which ensure the perceptual robustness of the algorithm. Geometric invariant distance is calculated between the Hu invariant moment vector and the maximum singular value vector to construct hash so that the algorithm reaches good classification performance between robustness and

uniqueness. Apart from that, random selection of image blocks and encrypting the hash sequence with the chaotic encryption make sure the high safety performance. The simulation experiment analysis indicates that the proposed perceptual hash has good comprehensive performance.

With the growing popularity of smart terminal applications such as mobile phones and tablets, a large number of screen content images have been generated. Therefore, we can further study the image quality evaluation hashing algorithm for screen content in the future. In addition, the computing and storage performance of mobile terminal equipments is limited. The current mainstream image hashing algorithms have limitations when applied to mobile devices. Therefore, the next step is to study the lightweight image hashing algorithm for mobile device applications.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Anhui Provincial Key Research and Development Plan (201904a05020091), the Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province (KJ2021A1030), the Key Scientific Research Projects of Chaohu University (XLZ-202108), Special Support Plan for Innovation and Entrepreneurship Leaders in Anhui Province, the Provincial Humanities and Social Science Research Program of Higher Education Institutions of Anhui Province (SK2021A0613), and the Key Scientific Research Projects of Chaohu University (XLZ-202104).

## References

[1] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2014.

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

[3] Z. Zhou, Q. M. J. Wu, S. Wan, W. Sun, and X. Sun, "Integrating SIFT and CNN feature matching for partial-duplicate image detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 593–604, 2020.

[4] W. Qi, B. Yue, C. Wangdu et al., "An overview on digital content watermarking," in *Signal and Information Processing, Networking and Computers. Lecture Notes in Electrical Engineering*, J. Sun, Y. Wang, M. Huo, and L. Xu, Eds., vol. 917, 2023.

[5] C.-M. Pun, C.-P. Yan, and X.-C. Yuan, "Robust image hashing using progressive feature selection for tampering detection," *Multimedia Tools and Applications*, vol. 77, no. 10, Article ID 11633, 2018.

[6] X. Wang, X. Zhou, Q. Zhang, B. Xu, and J. Xue, "Image alignment based perceptual image hash for content authentication," *Signal Processing: Image Communication*, vol. 80, Article ID 115642, 2020.

[7] L. Du, A. T. Ho, and R. Cong, "Perceptual hashing for image authentication: a survey," *Signal Processing: Image Communication*, vol. 81, Article ID 115713, 2020.

[8] Q. Li, X. Tian, W. W. Y. Ng, and S. Kwong, "Recent development of hashing-based image retrieval in non-stationary environments," *Int. J. Mach. Learn. & Cyber.*vol. 13, no. 12, pp. 3867–3886, 2022.

[9] X. Yang, Z. Cai, T. Jin, Z. Tang, and S. Gao, "A three-phase search approach with dynamic population size for solving the maximally diverse grouping problem," *European Journal of Operational Research*, vol. 302, no. 3, pp. 925–953, 2022.

[10] Bo Li, J. Xu, T. Jin, and Y. Shu, "Piecewise parameterization for multifactor uncertain system and uncertain inventory-promotion optimization," *Knowledge-Based Systems*, vol. 255, Article ID 109683, 2022.

[11] Y. Shu and Bo Li, "Stability analysis for uncertain nonlinear switched systems with infinite-time domain," *Fuzzy Optimization and Decision Making*, vol. 21, no. 3, pp. 405–428, 2022.

[12] B. Li, R. Zhang, T. Jin, and Y. Shu, "Parametric approximate optimal control of uncertain differential game with application to counter terror," *Chaos, Solitons & Fractals*, vol. 146, Article ID 110940, 2021.

[13] Q. Liu, T. Jin, M. Zhu, C. Tian, F. Li, and D. Jiang, "Uncertain currency option pricing based on the fractional differential equation in the caputo sense," *Fractal Fract*, vol. 6, no. 8, p. 407, 2022.

[14] Z. Tang, Z. Huang, H. Yao, X. 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.

[15] S. M. Abdullahi, H. Wang, and T. Li, "Fractal coding-based robust and alignment-free fingerprint image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2587–2601, 2020.

[16] M. Paul, R. K. Karsh, and F. Ahmed Talukdar, "Image hashing based on shape context and speeded up robust features (SURF)," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 464–468, London, UK, April 2019.

[17] S. P. Singh, G. Bhatnagar, and A. K. Singh, "A new robust reference image hashing system," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2211–2225, 2022.

[18] J. Ouyang, X. Wen, J. Liu, and J. Chen, "Robust hashing based on quaternion Zernike moments for image authentication," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 4, pp. 1–13, 2016.

[19] Z. Huang and S. Liu, "Perceptual hashing with visual content understanding for reduced-reference screen content image quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2808–2823, 2021.

[20] Z. Su, L. Yao, J. Mei, L. Zhou, and W. Li, "Learning to hash for personalized image authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1648–1660, 2021.

[21] S. Liu and Z. Huang, "Efficient image hashing with geometric invariant vector distance for copy detection," *ACM*

*Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 4, pp. 1–22, 2019.

[22] Z. Tang, H. Zhang, C.-M. Pun, M. Yu, C. Yu, and X. Zhang, "Robust image hashing with visual attention model and invariant moments," *IET Image Processing*, vol. 14, no. 5, pp. 901–908, 2020.

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

[24] Z. Tang, M. Yu, H. Yao, H. Zhang, C. Yu, and X. Q. Zhang, "Robust image hashing with singular values of quaternion SVD," *The Computer Journal*, vol. 64, no. 11, pp. 1656–1671, 2021.

[25] 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.

[26] X. Liang, Z. Tang, X. Xie, J. Wu, and X. Zhang, "Robust and fast image hashing with two-dimensional PCA," *Multimedia Systems*, vol. 27, no. 3, pp. 389–401, 2021.

[27] Y. Li, D. Wang, and L. Tang, "Robust and secure image fingerprinting learned by neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 362–375, 2020.

[28] C. Qin, E. Liu, G. Feng, and X. Zhang, "Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4523–4537, 2021.

[29] 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.

[30] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[31] R. Franzen, "Kodak Lossless True Color Image Suite," 2022, http://r0k.us/graphics/kodak/.

[32] G. Schaefer and M. Stich, "UCID.Anuncompressed colour image database," *Proceedings of SPIE*, vol. 5307, pp. 472–480, 2004.

[33] F. A. P. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Processing Magazine*, vol. 17, pp. 58–64, 2000.

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

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