algorithms

MDPI

Article

An Image Hashing Algorithm for Authentication with Multi-Attack Reference Generation and Adaptive Thresholding

Ling Du [1], Zehong He [1], Yijing Wang [1], Xiaochao Wang [2,*] and Anthony T. S. Ho [3,4]

[1] School of Computer Science and Technology, Tiangong University, Tianjin 300387, China; duling@tiangong.edu.cn (L.D.); 1831125468@tiangong.edu.cn (Z.H.); 1831125506@tiangong.edu.cn (Y.W.)
[2] School of Mathematical Sciences, Tiangong University, Tianjin 300387, China
[3] Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK; a.ho@surrey.ac.uk
[4] School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300457, China
* Correspondence: wangxiaochao@tiangong.edu.cn

Received: 18 August 2020; Accepted: 3 September 2020; Published: 8 September 2020

Abstract: Image hashing-based authentication methods have been widely studied with continuous advancements owing to the speed and memory efficiency. However, reference hash generation and threshold setting, which are used for similarity measures between original images and corresponding distorted version, are important but less considered by most of existing models. In this paper, we propose an image hashing method based on multi-attack reference generation and adaptive thresholding for image authentication. We propose to build the prior information set based on the help of multiple virtual prior attacks, and present a multi-attack reference generation method based on hashing clusters. The perceptual hashing algorithm was applied to the reference/queried image to obtain the hashing codes for authentication. Furthermore, we introduce the concept of adaptive thresholding to account for variations in hashing distance. Extensive experiments on benchmark datasets have validated the effectiveness of our proposed method.

Keywords: reference hashing; adaptive thresholding; image authentication

1. Introduction

With the aid of sophisticated photoediting software, multimedia content authentication is becoming increasingly prominent. Images edited by Photoshop may mislead people and cause social crises of confidence. In recent years, image manipulation has received a lot of criticism for its use in altering the appearance of image content to the point of making it unrealistic. Hence, tampering detection, a scheme that identifies the integrity and authenticity of the digital multimedia data, has emerged as an important research topic. Perceptual image hashing [1–4] supports image content authentication by representing the semantic content in a compact signature, which should be sensitive to content altering modifications but robust against content preserving manipulations such as blur, noise and illumination correction [5–7].

A perceptual image hashing system generally consists of three pipeline stages: the pre-processing stage, the hashing generation stage and the decision making stage. The major purpose of pre-processing is to enhance the robustness of features by preventing the effects of some distortions. After that, the reference hashes are generated and transmitted through a secure channel. For the test image, the same perceptual hash process will apply to the queried image to be authenticated. After the image hashing is generated, the task of image authentication can be validated by the decision making stage. The reference hash will be compared with image hashes in the test database for content

authentication based on the selected distance metric, such as Hamming distance. Currently the majority of perceptual hashing algorithms for authentication application can roughly be divided into the five categories: invariant feature transform-based methods [8–13], local feature points-based schemes [14–23], dimension reduction-based hashing [24–29], statistical features-based hashing [30–35] and leaning-based hashing [36–39].

For the decision making stage of perceptual hashing-based image authentication framework, only a few studies have been devoted to the reference generation and threshold selection. For reference hashing generation, Lv et al. [36] proposed obtaining an optimal estimate of the hash centroid using kernel density estimation (KDE). In this method, the centroid was obtained as the value which yields the maximum estimated distribution. Its major drawbacks are that the binary codes are obtained by using a data independent method. Since the hashing generation is independent of the data distribution, data independent hashing methods may not consider the characteristics of data distribution in hashing generation. Currently, more researchers are beginning to focus on the data dependent methods with learning for image tamper detection. Data dependent methods with learning [40–43] can be trained to optimally fit data distributions and specific objective functions, which produce better hashing codes to preserve the local similarity. In our previous work [44], we proposed a reference hashing method based on clustering. This algorithm makes the observation that the hashes of the original image actually not be the centroid of its cluster set. Therefore, how to learn the reference hashing code for solving multimedia security problems is an important topic for current research. As for authentication decision making, the simple way is to use threshold-based classifiers. Actually, perceptual differences under the image manipulations are often encountered when information is provided by different textural images. Traditional authentication tasks aim to identify the tampered results from distance values among different image codes. In this kind of task, the threshold is regarded as a fixed value. However, in a number of real-world cases, the objective truth cannot be identified by one fixed threshold for any image.

In this paper, we extend our previous work [44] and propose an image hashing algorithm framework for authentication with multi-attack reference generation and adaptive thresholding. According to the requirement of authentication application, we propose to build the prior information set based on the help of multiple virtual prior attacks, which is produced by applying virtual prior distortions and attacks on the original images. Differently from the traditional image authentication task, we address this uncertainty and introduce the concept of adaptive thresholding to account for variations in hashing distance. The main difference here is that a different threshold value is computed for each image. This technique provides more robustness to changes in image manipulations. We propose a data dependent semi-supervised image authentication scheme by using an attack-specific, adaptive threshold to generate a hashing code. This threshold tag is embedded in the hashing code transmission which can be reliably extracted at the receiver. The framework of our algorithm is shown in Figure 1. We firstly introduce the proposed multi-attack reference hashing algorithm. Then, we describe how original reference images were generated for experiments. After that, the perceptual hashing process was applied to the reference/queried image to be authenticated, so as to obtain the hashing codes. Finally, the reference hashes were compared with queried image hashes in the test database for content authentication.

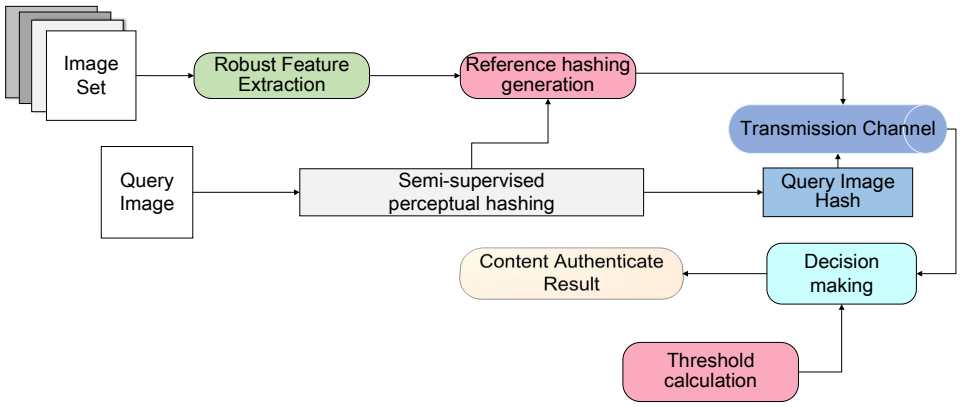


Figure 1. Block diagram of the proposed algorithm.

2. Problem Statement and Contributions

Authentication is an important issue of multimedia data protection; it makes possible to trace the author of the multimedia data and to allow the determination of whether an original multimedia data content was altered in any way from the time of its recording. The hash value is a compact abstract of the content. We can re-generate a hash value from the received content, and compare it with the original hash value. If they match, the content is considered as authentic. In the proposed algorithm, we aim to compute the common hashing function $h_k(.)$ for image authentication work. Let $D(.,.)$ indicate a decision making function for comparing two hash values. For given thresholds τ, the perceptual image hashing for tamper detection should satisfy the following criteria. If two images $\mathbf{x}$ and $\mathbf{y}$ are perceptually similar, their corresponding hashes need to be highly correlated, i.e., $D(h_k(\mathbf{x}), h_k(\mathbf{y})) < \tau$. Otherwise, If z is the tampered image from x, we should have $D(h_k(\mathbf{x}), h_k(\mathbf{z})) > \tau$.

The main contributions can be summarized as follows:

(1) We propose building the prior information set based on the help of multiple virtual prior attacks, which we did by applying virtual prior distortions and attacks to the original images. On the basis of said prior image set we aimed to infer the clustering centroids for reference hashing generation, which is used for a similarity measure.

(2) We effectively exploited the semi-supervised information into the perceptual image hashing learning. Instead of determining metric distance on training results, we explored the hashing distance for thresholding by considering the effect on different images.

(3) In order to account for variations in exacted features of different images, we took into account the pairwise variations among different originally-received image pairs. Those adaptive thresholding improvements maximally discriminate the malicious tampering from content-preserving operations, leading to an excellent tamper detection rate.

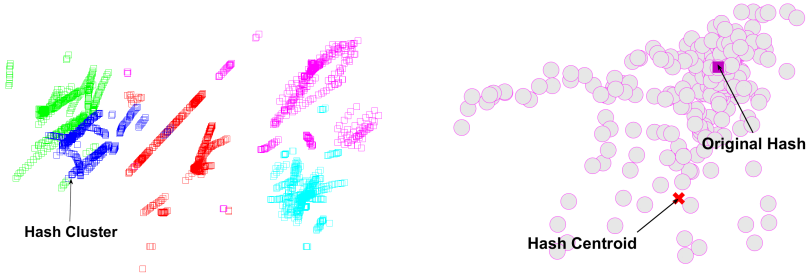
3. Proposed Method

3.1. Multi-Attack Reference Hashing

Currently, most image hashing method take the original image as the reference. However, the image hashes arising from the original image may not be the hash centroid of the distorted copies. As shown in Figure 2a, we applied 15 classes of attacks on five original images and represent their hashes in 2-dimensional space for both the original images and their distorted copies. From Figure 2a, we can observe five clusters in the hashing space. From Figure 2b by zooming into one hash cluster, we note an observation that the hashes of the original image actually may not be the centroid of its cluster.

For l original images in the dataset, we apply V type content preserving attacks with different types of parameter settings to generate simulated distorted copies. Let us denote the feature matrix of attacked instances in set Ψ_v as $\mathbf{X}^v \in \mathbb{R}^{m \times t}$. Here, $v = 1, 2, ..., V$, m is the dimensionality of data feature

and t is the number of instances for attack v. Finally, we get the feature matrices for the total n instance as $\mathbf{X} = \{\mathbf{X}^1, ..., \mathbf{X}^V\}$, and here $n = tV$. Note that the feature matrices are normalized to zero-centered.



(**a**) multiple hash clusters (**b**) a single hash cluster

Figure 2. The examples of hash clusters.

By considering the total reconstruction errors of all the training objects, we have the following minimization problem in a matrix form, which jointly exploits the information from various content preserving multi-attack data.

$$J_1(\tilde{\mathbf{U}}, \tilde{\mathbf{X}}) = \alpha(||\tilde{\mathbf{X}} - \tilde{\mathbf{U}}\mathbf{X}||_F^2 + \beta||\tilde{\mathbf{U}}||_F^2), \tag{1}$$

where $\tilde{\mathbf{X}}$ is the shared latent multi-attack feature representation. The matrix $\tilde{\mathbf{U}}$ can be viewed as the basis matrix, which maps the input multi-attack features onto the corresponding latent features. Parameter α, β is a nonnegative weighting vector to balance the significance.

From the information-theoretic point of view, the variance over all data is measured, and taken as a regularization term:

$$J_2(\tilde{\mathbf{U}}) = \gamma||\tilde{\mathbf{U}}\mathbf{X}||_F^2, \tag{2}$$

where γ is a nonnegative constant parameter.

The image reference for authentication is actually an infinite clustering problem. The reference is usually generated based on the cluster centroid image. Therefore, we also consider keeping the cluster structures. We formulate this objective function as:

$$J_3(\mathbf{C}, \mathbf{G}) = \lambda||\tilde{\mathbf{X}} - \mathbf{C}\mathbf{G}||_F^2, \tag{3}$$

where $\mathbf{C} \in \mathbb{R}^{k \times l}$ and $\mathbf{G} \in \{0, 1\}^{l \times n}$ are the clustering centroid and indicator.

Finally, the formulation can be written as:

$$\min_{\tilde{\mathbf{U}}, \tilde{\mathbf{X}}, \mathbf{C}, \mathbf{G}} \alpha||\tilde{\mathbf{X}} - \tilde{\mathbf{U}}\mathbf{X}||_F^2 + \beta||\tilde{\mathbf{U}}||_F^2 - \gamma||\tilde{\mathbf{U}}\mathbf{X}||_F^2 + \lambda||\tilde{\mathbf{X}} - \mathbf{C}\mathbf{G}||_F^2. \tag{4}$$

Our objective function simultaneously learns the feature representations $\tilde{\mathbf{X}}$ and finds the mapping matrix $\tilde{\mathbf{U}}$, the cluster centroid $\mathbf{C}$ and indicator $\mathbf{G}$. The iterative optimization algorithm is as follows.

Fixing all variables but **optimize** $\tilde{\mathbf{U}}$: The optimization problem (Equation (4)) reduces to:

$$min\, J(\tilde{\mathbf{U}}^v) = \alpha||\tilde{\mathbf{X}} - \tilde{\mathbf{U}}\mathbf{X}||_F^2 + \beta||\tilde{\mathbf{U}}||_F^2 - \gamma\mathbf{tr}(\tilde{\mathbf{U}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\tilde{\mathbf{U}}^{\mathrm{T}}). \tag{5}$$

By setting the derivation $\frac{\partial J(\tilde{\mathbf{U}})}{\partial \tilde{\mathbf{U}}}$=0, we have:

$$\tilde{\mathbf{U}} = \tilde{\mathbf{X}}\mathbf{X}^{\mathrm{T}}((\alpha - \gamma)\mathbf{X}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I})^{-1}. \tag{6}$$

Fix all variables but **optimize** $\tilde{\mathbf{X}}$: Similarly, we solve the following optimization problem:

$$min\, F(\tilde{\mathbf{X}}) = \alpha||\tilde{\mathbf{X}} - \tilde{\mathbf{U}}\mathbf{X}||_F^2 + \lambda||\tilde{\mathbf{X}} - \mathbf{C}\mathbf{G}||_F^2, \tag{7}$$

which has a closed-form optimal solution:

$$\tilde{\mathbf{X}} = \alpha\tilde{\mathbf{U}}\mathbf{X} + \lambda\mathbf{C}\mathbf{G}. \tag{8}$$

Fix all variables but **C** and **G**: For the cluster centroid **C** and indicator **G**, we obtain the following problem:

$$\min_{\mathbf{C},\mathbf{G}} ||\tilde{\mathbf{X}} - \mathbf{C}\mathbf{G}||_F^2. \tag{9}$$

Inspired by the optimization algorithm ADPLM (adaptive discrete proximal linear method) [45], we initialize $\mathbf{C} = \tilde{\mathbf{X}}\mathbf{G}^{\mathrm{T}}$ and update **C** as follows:

$$\mathbf{C}^{p+1} = \mathbf{C}^p - \frac{1}{\mu} \bigtriangledown \Gamma(\mathbf{C}^p), \tag{10}$$

where $\Gamma(\mathbf{C}^p) = ||\mathbf{B} - \mathbf{C}\mathbf{G}||_F^2 + \rho||\mathbf{C}^{\mathrm{T}}1||$, $\rho = 0.001$, $p = 1, 2, ...5$ denote the p-th iteration.

The indicator matrix **G** at indices (i, j) is obtained by:

$$g_{i,j}^{p+1} = \begin{cases} 1 & j = arg\min_s H(b_i, c_s^{p+1}) \\ 0 & otherwise \end{cases}, \tag{11}$$

where $H(b_i, c_s)$ is the distance between the i-th feature codes x_i and the s-th cluster centroid c_s.

After we infer the cluster centroid **C** and the multi-attack feature representations $\tilde{\mathbf{X}}$, the corresponding l reference images are generated. The basic idea is to compare the hashing distances among the nearest content, preserving the attacked neighbors of each original image and corresponding cluster centroid.

3.2. Semi-Supervised Hashing Code Learning

For the reference and received images, we use the semi-supervised learning algorithm for hashing code generation and image authentication. Firstly, all the input image is converted to a normalized size 256×256 by using the bi-linear interpolation. The resizing operation makes our hashing robust against image rescaling. Then, the Gaussian low-pass filter is used to blur the resized image, which can reduce the influences of high-frequent components on the image, such as noise contamination or filtering. Let $F(i, j)$ be the element in the i-th row and the j-th column of the convolution mask. It is calculated by

$$F(i, j) = \frac{F^{(1)}(x, y)}{\sum_i \sum_j F^{(1)}(x, y)}, \tag{12}$$

in which $F^{(1)}(x, y)$ is defined as

$$F^{(1)}(x, y) = e^{\frac{-(i^2+j^2)}{2\sigma^2}}, \tag{13}$$

where σ is the standard deviation of all elements in the convolution mask.

Next, the RGB color image is converted into the CIE LAB space and the image is represented by the L component. The reason is that the L component closely matches human perception of lightness. The RGB color image is firstly converted into the XYZ color space by the following formula:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{14}$$

where R, G, and B are the red, green and blue components of the color pixel. We convert it into the CIE LAB space by the following equation:

$$\begin{aligned} L &= 116f(Y/Y_w) - 16 \\ A &= 500f[(X/X_w) - f(Y/Y_w)], \\ B &= 200f[(X/X_w) - f(Z/Z_w)] \end{aligned} \tag{15}$$

where $X_w = 0.950456$, $Y_w = 1.0$ and $Z_w = 1.088754$ are the CIE XYZ tristimulus values of the reference white point, and $f(t)$ is determined by:

$$f(t) = \begin{cases} t^{1/3}, & \text{if } t > 0.008856 \\ 7.787t + 16/116, & otherwise \end{cases}. \tag{16}$$

Figure 3 illustrates an example of the preprocessing.

(**a**) Input image (**b**) Resized image (**c**) Blurred result (**d**) L component

Figure 3. An example of preprocessing.

Let us say we have N images in our training set. Select L images as labeled images, $L \ll N$. The features of a single image are expressed as $\mathbf{x} \in \mathbb{R}^M$, where M is the extracted feature length. The features of all images are represented as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where $\mathbf{X} \in \mathbb{R}^{M \times N}$. The features of labeled images are represented as $\mathbf{X} \in \mathbb{R}^{M \times L}$. Note that these feature matrices are normalized to zero-centered. The goal of our algorithm is to learn hash functions that map $\mathbf{X} \in \mathbb{R}^{M \times N}$ to a compact representation $\mathbf{H} \in \mathbb{R}^{K \times N}$ in a low-dimensional Hamming space, where K is the digits length. Our hash function is defined as:

$$\mathbf{H} = \mathbf{W}^{\mathrm{T}}\mathbf{X}. \tag{17}$$

The hash function of a single image is defined as:

$$\mathbf{h_i} = \mathbf{W}^{\mathrm{T}}\mathbf{x}_i. \tag{18}$$

In order to learn a $\mathbf{W}$ that is simultaneously maximizing the empirical accuracy on the labeled image and the variance of hash bits over all images, the empirical accuracy on the labeled image is defined as:

$$P_1(\mathbf{W}) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \mathbf{E}_{ij}\mathbf{h}_i\mathbf{h}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \mathbf{E}_{ij}\mathbf{h}_i\mathbf{h}_j, \tag{19}$$

where matrix $\mathbf{E}$ is the classification of marked image pairs, as follows:

$$\mathbf{E}(i,j) = \begin{cases} 1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ -1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} \\ 0 & otherwise \end{cases}, \tag{20}$$

Specifically, a pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ is denoted as a perceptually similar pair when the two images are the same images or the attacked images of a same image, and a pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}$ is denoted as a perceptually different pair when the two images are different images or when one suffered from malicious manipulations or perceptually significant attacks.

Equation (19) can also be represented as:

$$P_1(\mathbf{W}) = \frac{1}{2}\mathbf{tr}\{(\mathbf{W}^{\mathrm{T}}\mathbf{X_n})\,\mathbf{E}\,(\mathbf{W}^{\mathrm{T}}\mathbf{X_n})^{\mathrm{T}}\}. \tag{21}$$

This relaxation is quite intuitive. That is, the similar images are desired to not only have the same sign but also large projection magnitudes, while the projections for dissimilar images not only have different signs but also are as different as possible.

Moreover, to maximize the amount of information per hash bit, we want to calculate the maximum variance of all hash bits of all images and use it as a regularization term of the hash function.

$$V(\mathbf{W}) = \sum_{i=1}^{N} var(\mathbf{h_i}) = \sum_{N} var(\mathbf{W}^{\mathrm{T}}\mathbf{x}_i). \tag{22}$$

Due to the indifferentiability of the above function, it is difficult to calculate its extreme value. However, the maximum variance of the hash function is the lower bound of the scale variance of the projected data, so the information theoretic regularization is represented as:

$$P_2(\mathbf{W}) = \frac{1}{2}\mathbf{tr}\{(\mathbf{W}^{\mathrm{T}}\mathbf{X})(\mathbf{W}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}}\}. \tag{23}$$

Finally, the overall semi-supervised objective function combines the relaxed empirical fitness term from Equation (21) and the regularization term from Equation (23).

$$P(\mathbf{W}) = P_1(\mathbf{W}) + \eta P_2(\mathbf{W}) = \frac{1}{2}\mathrm{tr}\{\mathbf{W}^{\mathrm{T}}(\mathbf{X_n}\mathbf{E}\mathbf{X_n}^{\mathrm{T}} + \eta\mathbf{X}\mathbf{X}^{\mathrm{T}})\mathbf{W}\}, \tag{24}$$

where $\eta = 0.25$ is a tradeoff parameter. The optimization problem is as follows:

$$\underset{\mathbf{W}}{max} P(\mathbf{W}) \quad s.t.\, \mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{I}, \tag{25}$$

where the constraint $\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{I}$ makes the projection directions orthogonal. We learn the optimal projection $\mathbf{W}$ that is obtained by means of eigenvalue decomposition of matrix $\mathbf{M}$.

3.3. Adaptive Thresholds-Based Decision Making

To measure the similarity between hashes of original and attacked/tampered images, the metric distance between two hashing code is calculated by:

$$d(\mathbf{h_1}, \mathbf{h_2}) = \| \frac{\mathbf{h_1} - \mathbf{h_2}}{2\sqrt{\|\mathbf{h_1}\|\|\mathbf{h_2}\|}} \|, \tag{26}$$

where $\mathbf{h_1}$ and $\mathbf{h_2}$ are two image hashes. In general, the more similar the images, the smaller the distance. The greater the difference, the greater the distance.

Then, the threshold T is defined to judge whether the image is a similar image or a tampered image.

$$\begin{cases} Similar\ images\ pair, & \text{if } (d \le T) \\ Tampered\ images\ pair, & \text{if } (d > T) \end{cases}. \tag{27}$$

If the distance is less than a given threshold, the two images are judged as visually identical images. Otherwise, they are judged as distinct images.

Traditional image tamper detection algorithms take a fixed value as the threshold to judge similar images/tampered images. However, due to the different characteristics among images, some images cannot be correctly judged by the fixed threshold value. In our adaptive thresholds algorithm, we firstly find the maximum value for the distance value of the similar images and the minimum value for the

distance value of the tampered images. In order to prevent the two values from being too extreme, we set the following limits:

$$\begin{array}{ll} dist_{min} = \max(dist1) & s.t.\ (dist_{min} - median(dist1)) > \psi \\ dist_{max} = \min(dist2) & s.t.\ dist_{max} > \xi \end{array}, \tag{28}$$

where $dist1$ is the distance between the similar image and the original image; $dist2$ is the distance between the tampered image and the original image. ψ and ξ are two constants set experimentally. Then, the resulting maximum and minimum values are compared with fixed thresholds:

$$\widetilde{\tau} = \begin{cases} \tau, & \text{if } (dist_{min} < \tau < dist_{max}) \\ dist_{max} - (dist_{max} - dist_{min})/3, & \text{if } (dist_{min} < dist_{max} \leq \tau) \\ dist_{min} + (dist_{max} - dist_{min})/3, & \text{if } (\tau \leq dist_{min} < dist_{max}) \\ (dist_{min} + dist_{max})/2, & otherwise \end{cases}, \tag{29}$$

where τ is a fixed threshold obtained experimentally, $\widetilde{\tau}$ is the adaptive threshold suitable for this image. Then, all images have their own thresholds, which are represented as:

$$\widetilde{T} = [\widetilde{\tau}_1, \widetilde{\tau}_2, ..., \widetilde{\tau}_n]. \tag{30}$$

Finally, we put the adaptive threshold at the top of the hash code and transfer it along with the hash code. Thus, the final hash code is represented as:

$$\widetilde{\mathbf{h_i}} = [\widetilde{\tau}_i, \mathbf{h_i}]. \tag{31}$$

4. Experiments

4.1. Data

Our experiments were carried out on two real-world datasets. The first came from the **CASIA** [46], which contains 918 image pairs, including 384 × 256 real images and corresponding distorted images with different texture characteristics. The other one was **RTD** [47,48], which contains 220 real images and corresponding distorted images with resolution 1920 × 1080.

To ensure that the images of the training set were different from the images of the testing set, we selected 301 non-repetitive original images and their corresponding tampered images to generate 66,231 images as our training data. Furthermore, 10,000 images were randomly selected from 66,231 images as a labeled subset. We adopted 226 repetitive original images and their corresponding set of tampered images to determine the threshold value of each image. The remaining images in CASIA and RTD datasets were used to test performance.

4.2. Baselines

We compared our proposed algorithm with a number of baselines. In particular, we compared it with:

Wavelet-based image hashing [49]: It is an invariant feature transform-based method, which develops an image hash from the various sub-bands in a wavelet decomposition of the image and makes it convenient to transform from the space-time domain to the frequency.

SVD-based image hashing [24]: It belongs to dimension reduction-based hashing and it uses spectral matrix invariants as embodied by singular value decomposition. The invariant features based on matrix decomposition show good robustness against noise addition, blurring and compressing attacks.

RPIVD-based image hashing [30]: It incorporates ring partition and invariant vector distance into image hashing by calculating the images statistics. The statistical information of the images includes: mean, variance, standard deviation, kurtosis, etc.

Quaternion-based image hashing [12]: This method considers multiple features, and constructs a quaternion image to implement a quaternion Fourier transform for hashing generation.

4.3. Perceptual Robustness

To validate the perceptual robustness of proposed algorithm, we applied twelve types of content preserving operations: (a) Gaussian noise addition with the variance of 0.005. (b) Salt and pepper noise addition with a density of 0.005. (c) Gaussian blurring with the standard deviation of the filter 10. (d) Circular blurring with a radius of 2. (e) Motion blurring with the amount of the linear motion 3 and the angle of the motion blurring filter 45. (f) Average filtering with filter size of 5. (g) Median filtering with filter size of 5. (h) Wiener filtering with filter size of 5. (i) Image sharpening with the parameter alpha of 0.49. (j) Image scaling with the percentage 1.2. (k) Illumination correction with parameter gamma 1.18. (l) JPEG compression with quality factor 20.

We extracted the reference hashing code based on the original image (ORH) and our proposed multi-attack reference hashing (MRH). For the content-preserving distorted images, we calculated the corresponding distances between reference hashing codes and content-preserving images' hashing codes. The statistical results under different attacks are presented in Table 1. Just as shown, the hashing distances for the four baseline methods were small enough. In our experiments, we set the threshold $\tau = 0.12$ to distinguish the similar images and forgery images from the CASIA dataset for the PRIVD method. Similarly, for the other three methods, we set the thresholds as 1.2, 0.0012 and 0.008 correspondingly for their best results.

Table 1. Hashing distances under different content-preserving manipulations.

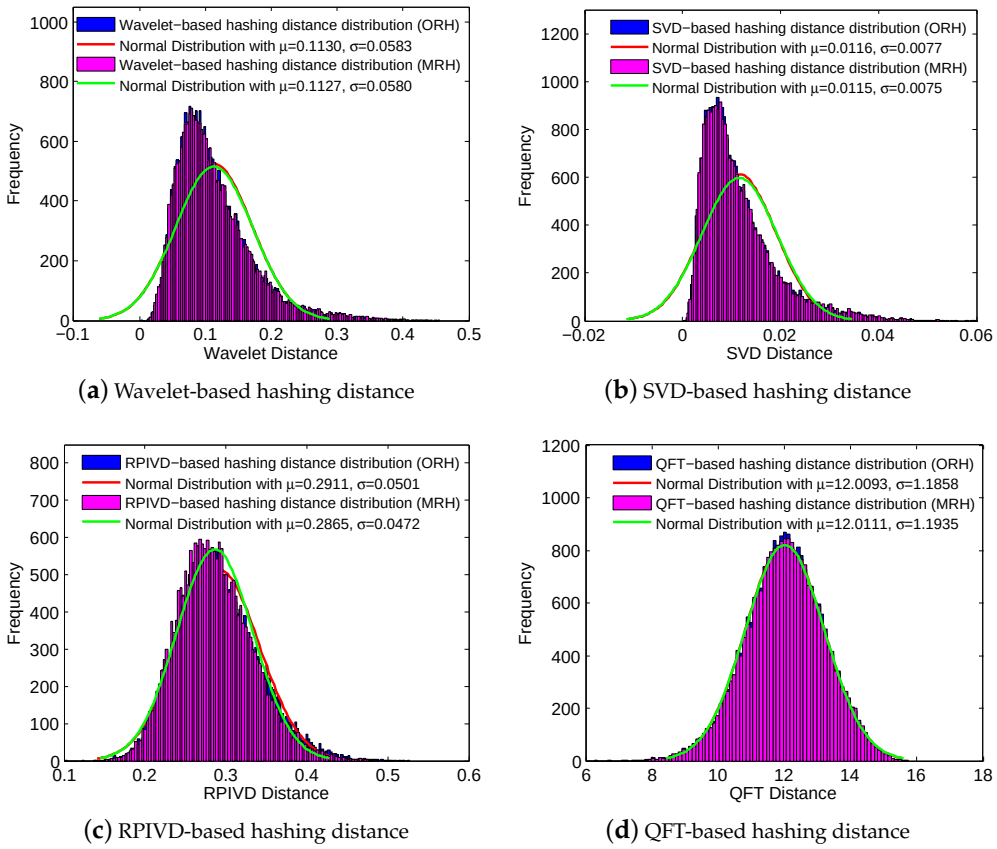
Method	Manipulation	ORH			MRH		
		Max	**Min**	**Mean**	**Max**	**Min**	**Mean**
Wavelet	Gaussian noise	0.02828	0.00015	0.00197	0.02847	0.00014	0.00196
	Salt&Pepper	0.01918	0.00021	0.00252	0.01918	0.00024	0.00251
	Gaussian blurring	0.00038	0.00005	0.00017	0.00067	0.00006	0.00019
	Circular blurring	0.00048	0.00006	0.00022	0.00069	0.00006	0.00021
	Motion blurring	0.00034	0.00006	0.00015	0.00065	0.00005	0.00016
	Average filtering	0.00071	0.00007	0.00033	0.00071	0.00009	0.00030
	Median filtering	0.00704	0.00006	0.00099	0.00753	0.00007	0.00099
	Wiener filtering	0.00101	0.00008	0.00028	0.00087	0.00008	0.00028
	Image sharpening	0.00906	0.00009	0.00115	0.00906	0.00010	0.00114
	Image scaling	0.00039	0.00005	0.00013	0.00064	0.00006	0.00018
	Illumination correction	0.08458	0.00447	0.02759	0.08458	0.00443	0.02757
	JPEG compression	0.00143	0.00009	0.00026	0.00275	0.00013	0.00051
SVD	Gaussian noise	0.00616	0.00007	0.00031	0.00616	0.00007	0.00030
	Salt&Pepper	0.00339	0.00008	0.00034	0.00338	0.00007	0.00033
	Gaussian blurring	0.00017	0.00007	0.00010	0.00113	0.00007	0.00011
	Circular blurring	0.00018	0.00006	0.00010	0.00114	0.00006	0.00011
	Motion blurring	0.00017	0.00007	0.00010	0.00113	0.00006	0.00011
	Average filtering	0.00025	0.00007	0.00011	0.00111	0.00006	0.00012
	Median filtering	0.00166	0.00007	0.00015	0.00190	0.00007	0.00016
	Wiener filtering	0.00035	0.00005	0.00011	0.00113	0.00007	0.00012
	Image sharpening	0.00104	0.00007	0.00018	0.00099	0.00007	0.00018
	Image scaling	0.00016	0.00007	0.00010	0.00114	0.00007	0.00011
	Illumination correction	0.00662	0.00014	0.00149	0.00674	0.00014	0.00150
	JPEG compression	0.00031	0.00007	0.00010	0.00053	0.00008	0.00012

Table 1. *Cont.*

Method	Manipulation	ORH			MRH		
		Max	**Min**	**Mean**	**Max**	**Min**	**Mean**
RPIVD	Gaussian noise	0.25827	0.00864	0.03086	0.29081	0.01115	0.03234
	Salt&Pepper	0.22855	0.01131	0.02993	0.25789	0.01191	0.03033
	Gaussian blurring	0.03560	0.00411	0.01471	0.14023	0.00545	0.01786
	Circular blurring	0.06126	0.00447	0.01713	0.13469	0.00565	0.01924
	Motion blurring	0.03570	0.00362	0.01432	0.18510	0.00473	0.01825
	Average filtering	0.07037	0.00543	0.02109	0.20190	0.00591	0.02237
	Median filtering	0.06126	0.00512	0.02234	0.18360	0.00625	0.02465
	Wiener filtering	0.07156	0.00421	0.01803	0.20421	0.00581	0.02041
	Image sharpening	0.06324	0.00609	0.02442	0.18283	0.00706	0.02765
	Image scaling	0.03311	0.00275	0.01154	0.18233	0.00381	0.01761
	Illumination correction	0.11616	0.00769	0.02864	0.20944	0.01047	0.02920
	JPEG compression	0.07037	0.00543	0.02109	0.06180	0.00707	0.02155
QFT	Gaussian noise	6.97151	0.13508	0.73563	6.30302	0.11636	0.60460
	Salt&Pepper	7.63719	0.16998	0.66200	7.50644	0.15073	0.63441
	Gaussian blurring	0.26237	0.00513	0.02519	0.10820	0.00318	0.01449
	Circular blurring	0.26529	0.00712	0.03163	0.17937	0.00460	0.02075
	Motion blurring	0.26408	0.00465	0.02286	0.10729	0.00300	0.01318
	Average filtering	0.30154	0.00976	0.04403	0.30719	0.00760	0.03263
	Median filtering	0.95120	0.03084	0.19822	0.87149	0.02706	0.19345
	Wiener filtering	0.64373	0.01746	0.08046	0.68851	0.01551	0.07616
	Image sharpening	6.55606	0.05188	1.52398	6.55596	0.05189	1.52398
	Image scaling	0.51083	0.04031	0.10067	0.52404	0.02800	0.09827
	Illumination correction	4.37001	0.27357	0.84280	4.36692	0.27348	0.84170
	JPEG compression	7.55523	0.13752	1.29158	13.1816	0.13585	1.46682

4.4. Discriminative Capability

The discriminative capability of a image hashing means that visually distinct images should have significantly different hashes. In other words, two images that are visually distinct should have a very low probability of generating similar hashes. Here, RTD dataset consisting of 220 different uncompressed color images was adopted to validate the discriminative capability of our proposed multi-attack reference hashing algorithm. We first extracted reference hashing codes for all 220 images in RTD and then calculated the hashing distance for each image with the other 219 images. Thus, we can finally obtained $220 \times (220-1)/2 = 24{,}090$ hashing distances. Figure 4 shows the distribution of these 24,090 hashing distances between hashing pairs with varying thresholds, where the abscissa is the hashing distance and the ordinate represents the frequency of hashing distance. It can be seen clearly from the histogram that the proposed method has good discriminative capability. For instance, we set $\tau = 0.12$ as the threshold on CASIA dataset when extracting the reference hashing by RPIVD method. The minimum value for hashing distance was 0.1389, which is above the threshold. The results show that the multi-attack reference hashing can replace the original image-based reference hashing with good discrimination.



(**a**) Wavelet-based hashing distance (**b**) SVD-based hashing distance

(**c**) RPIVD-based hashing distance (**d**) QFT-based hashing distance

Figure 4. Distribution of hashing distances between hashing pairs with varying thresholds.

4.5. Authentication Results

As the reference hashing performance for authentication, we compared the proposed multi-attack reference hashing (MRH) with original image-based reference hashing (ORH) on four baseline image hashing methods, i.e., wavelet-based image hashing, SVD-based image hashing, RPIVD-based image hashing and QFT-based image hashing, with twelve content-preserving operations. The results are shown in Tables 2 and 3. Note that higher values indicate better performance for all metrics. It was observed that the proposed MRH algorithm outperformed the ORH algorithm by a clear margin, irrespective of the content preserving operation and image datasets (RTD and CASIA). This is particularly evident for illumination correction. For instance, in contrast to original image-based reference hashing, the multi-attack reference hashing increased the AUC of illumination correction by 21.98% on the RTD image dataset when getting the reference hashing by wavelet, as shown in Table 2. For the QFT approach, the multi-attack reference hashing we proposed was more stable and outstanding than other corresponding reference hashings. Since the QFT robust image hashing technique is used to process the three channels of the color image, the chrominance information of the color image can be prevented from being lost and image features are more obvious. Therefore, the robustness of the multi-attack reference hashing is more able to resist geometric attacks and content preserving operations. For instance, the multi-attack reference hashing increased the precision of Gaussian noise by 3.28% on the RTD image.

Table 2. Comparisons between the original image-based reference hashing and the proposed multi-attack reference hashing (RTD dataset).

Manipulation	Wavelet				SVD				RPIVD				QFT			
	Precision	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**
Original image-based reference hashing																
Gaussian noise	0.6257	0.9500	0.7545	0.8442	0.8537	0.4773	0.6122	0.8501	0.8326	0.8211	0.8268	0.8991	0.8978	0.7591	0.8227	0.9241
Salt&Pepper	0.5485	0.9773	0.7026	0.8043	0.8537	0.4773	0.6122	0.8507	0.8806	0.8119	0.7449	0.9088	0.8851	0.7727	0.8252	0.9184
Gaussian blurring	1.0000	0.8727	0.9320	0.9866	1.0000	0.4409	0.6120	0.9874	1.0000	0.7465	0.8549	0.9557	1.0000	0.7227	0.8391	0.9948
Circular blurring	1.0000	0.8733	0.9346	0.9787	1.0000	0.4364	0.6076	0.9852	0.9821	0.7604	0.8571	0.9447	1.0000	0.7227	0.8391	0.9948
Motion blurring	1.0000	0.8727	0.9346	0.9787	1.0000	0.4273	0.5987	0.9868	1.0000	0.7477	0.8556	0.9572	1.0000	0.7227	0.8391	0.9949
Average filtering	1.0000	0.8864	0.9398	0.9665	1.0000	0.4318	0.6032	0.9790	0.9598	0.7661	0.8520	0.9351	1.0000	0.7227	0.8391	0.9948
Median filtering	0.6967	0.9500	0.8038	0.9012	0.9898	0.4409	0.6101	0.9544	0.9399	0.7890	0.8579	0.9212	1.0000	0.7409	0.8512	0.9721
Wiener filtering	0.9847	0.8773	0.9279	0.9713	1.0000	0.4318	0.6032	0.9822	0.9880	0.7569	0.8571	0.9427	1.0000	0.7227	0.8391	0.9950
Image sharpening	0.7178	0.9364	0.8126	0.8872	0.9709	0.4545	0.6192	0.9368	0.8980	0.8073	0.8502	0.9155	0.8851	0.8537	0.8537	0.8537
Image scaling	1.0000	0.8773	0.9346	0.9892	1.0000	0.4318	0.6032	0.9873	1.0000	0.7385	0.8496	0.9677	0.8851	0.7727	0.8252	0.9184
Illumination correction	0.5000	1.0000	0.6667	0.5593	0.5479	0.8318	0.6606	0.6754	0.9021	0.8028	0.8495	0.9073	0.6429	0.9000	0.7500	0.8498
JPEG compression	1.0000	0.4909	0.6585	0.9271	1.0000	0.4318	0.6032	0.9846	1.0000	0.3073	0.4702	0.9408	0.9273	0.6955	0.7948	0.9015
Multi-attack reference hashing																
Gaussian noise	0.8345	0.5273	0.6462	0.8465	0.8462	0.3000	0.4430	0.8846	0.9600	0.3303	0.4915	0.8948	0.9279	0.8773	0.9019	0.9588
Salt&Pepper	0.7619	0.5818	0.6598	0.8046	0.9507	0.6136	0.7459	0.9263	0.9706	0.3028	0.4615	0.9057	0.9500	0.6909	0.8000	0.9355
Gaussian blurring	1.0000	0.6000	0.7500	0.9955	1.0000	0.6045	0.7535	0.9904	0.9927	0.6415	0.7794	0.9880	1.0000	0.6818	0.8108	0.9952
Circular blurring	1.0000	0.4955	0.6626	0.9811	1.0000	0.6045	0.7535	0.9904	0.9926	0.6368	0.7759	0.9870	1.0000	0.6818	0.8108	0.9953
Motion blurring	1.0000	0.4909	0.6585	0.9849	1.0000	0.6000	0.7500	0.9955	0.9855	0.6415	0.7771	0.9857	1.0000	0.6818	0.8108	0.9952
Average filtering	1.0000	0.4955	0.6626	0.9709	1.0000	0.6091	0.7571	0.9955	0.9714	0.3119	0.4722	0.9270	1.0000	0.6818	0.8108	0.9952
Median filtering	0.9590	0.5318	0.6842	0.9013	0.9926	0.6091	0.7549	0.9803	1.0000	0.3211	0.4861	0.9258	1.0000	0.6818	0.8108	0.9809
Wiener filtering	1.0000	0.4909	0.6585	0.9703	1.0000	0.6045	0.7535	0.9901	0.9854	0.6368	0.7736	0.9858	1.0000	0.6864	0.8140	0.9950
Image sharpening	0.8986	0.5636	0.6927	0.8884	1.0000	0.2864	0.4452	0.9313	0.9722	0.3211	0.4828	0.9071	0.9167	0.7000	0.7938	0.9011
Image scaling	1.0000	0.4955	0.6626	0.9828	1.0000	0.6000	0.7500	0.9955	0.9855	0.6415	0.6415	0.9868	0.9494	0.6818	0.7937	0.9607
Illumination correction	0.5046	1.0000	0.6707	0.7791	0.6376	0.8318	0.7219	0.7848	0.9714	0.3119	0.4722	0.9062	0.7500	0.7909	0.7699	0.8405
JPEG compression	1.0000	0.4909	0.6585	0.9256	1.0000	0.6045	0.7535	0.9900	1.0000	0.3073	0.4702	0.9264	0.9403	0.8591	0.8979	0.9598

Table 3. Comparisons between original image-based reference hashing and the proposed multi-attack reference hashing (CASIA dataset).

Manipulation	Wavelet				SVD				RPIVD				QFT			
	Precision	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**
Original image-based reference hashing																
Gaussian noise	0.7451	0.6623	0.8010	0.7909	0.8385	0.7015	0.7639	0.8825	0.9782	0.6830	0.8044	0.9021	0.8802	0.8965	0.8883	0.9520
Salt&Pepper	0.8128	0.6481	0.7212	0.8307	0.8978	0.6983	0.7855	0.9164	0.9699	0.6329	0.7660	0.9282	0.8837	0.8856	0.8847	0.9572
Gaussian blurring	0.9694	0.5861	0.7305	0.9434	0.9937	0.6852	0.8811	0.9512	0.9502	0.6452	0.7685	0.8981	1.0000	0.8638	0.9269	0.9989
Circular blurring	0.9399	0.5959	0.7293	0.9526	0.9696	0.6939	0.8089	0.8274	0.8124	0.6856	0.7436	0.8467	1.0000	0.8638	0.9269	0.9989
Motion blurring	0.9745	0.5817	0.7285	0.9526	0.9952	0.6797	0.8078	0.9642	0.9827	0.6201	0.7604	0.9161	1.0000	0.8638	0.9269	0.9989
Average filtering	0.8786	0.6231	0.7291	0.8917	0.8728	0.7179	0.7878	0.8835	0.6562	0.7738	0.7101	0.7739	1.0000	0.8638	0.9269	0.9989
Median filtering	0.8838	0.6503	0.7307	0.8457	0.9269	0.7048	0.8007	0.9047	0.7296	0.7216	0.7256	0.8080	1.0000	0.8649	0.9276	0.9939
Wiener filtering	0.8997	0.6155	0.7309	0.9055	0.9485	0.7015	0.8065	0.9212	0.8227	0.6921	0.7539	0.8506	1.0000	0.8638	0.9269	0.9980
Image sharpening	0.7194	0.7197	0.7186	0.7878	0.8089	0.7702	0.7891	0.8656	0.6526	0.8268	0.7295	0.8014	0.6565	0.9390	0.7727	0.8653
Image scaling	0.9868	0.5719	0.7241	0.9640	0.9952	0.6808	0.8085	0.9672	0.9581	0.6234	0.7553	0.9180	1.0000	0.8627	0.9263	0.9986
Illumination correction	0.5008	0.9978	0.6669	0.6063	0.6256	0.8573	0.7233	0.7541	0.9941	0.5579	0.7147	0.9810	0.8854	0.9085	0.8968	0.9616
JPEG compression	1.0000	0.4909	0.6585	0.9271	0.9676	0.6830	0.8008	0.9580	0.9565	0.6495	0.7736	0.9076	0.7148	0.9281	0.8076	0.8861
Multi-attack reference hashing																
Gaussian noise	0.7604	0.6569	0.7049	0.7993	0.8647	0.6961	0.7713	0.8902	0.9429	0.8638	0.9016	0.9578	0.9130	0.8922	0.9025	0.9646
Salt&Pepper	0.8415	0.6362	0.7246	0.8407	0.9261	0.6961	0.7948	0.9202	0.9738	0.8497	0.9075	0.9693	0.8906	0.8954	0.8930	0.9614
Gaussian blurring	1.0000	0.5664	0.7232	0.9797	1.0000	0.6634	0.7976	0.9807	0.9584	0.8046	0.8748	0.9481	1.0000	0.8758	0.9338	0.9989
Circular blurring	0.9943	0.5708	0.7253	0.9624	0.9951	0.6645	0.7969	0.9644	0.8596	0.8155	0.8370	0.9081	1.0000	0.8758	0.9338	0.9989
Motion blurring	1.0000	0.5654	0.7223	0.9800	1.0000	0.6656	0.7992	0.9857	0.9867	0.8079	0.8884	0.9618	1.0000	0.8758	0.9338	0.9989
Average filtering	0.9451	0.6002	0.7342	0.9201	0.9574	0.6852	0.7987	0.9203	0.6915	0.8328	0.7556	0.8349	1.0000	0.8758	0.9338	0.9989
Median filtering	0.7954	0.6438	0.7116	0.8366	0.9077	0.6961	0.7879	0.9038	0.7795	0.8297	0.8038	0.8851	1.0000	0.8769	0.9344	0.9958
Wiener filtering	0.9818	0.5871	0.7348	0.9369	0.9842	0.6776	0.8026	0.9542	0.8659	0.8177	0.8411	0.9195	1.0000	0.8769	0.9344	0.9984
Image sharpening	0.7271	0.7081	0.7174	0.7958	0.7901	0.7789	0.7844	0.8581	0.6722	0.9292	0.7801	0.8982	0.6579	0.9434	0.7749	0.8685
Image scaling	0.9923	0.5599	0.7159	0.9521	0.9952	0.6754	0.8047	0.9657	0.9716	0.8210	0.8899	0.9640	1.0000	0.8780	0.9350	0.9988
Illumination correction	0.5008	0.9978	0.6669	0.6043	0.6003	0.8638	0.7084	0.7389	0.9973	0.8111	0.8946	0.9915	0.8843	0.9161	0.8999	0.9649
JPEG compression	0.9925	0.5763	0.7292	0.9420	0.9779	0.6754	0.7990	0.9537	0.9720	0.8368	0.8994	0.9627	0.7145	0.9270	0.8070	0.8859

For performance analysis, we took wavelet-based and SVD-based image hashing to extract features and used the semi-supervised method to train **W** for each content-preserving manipulations. The experimental results are summarized in Table 4. They show the probability of the true authentication capability of the proposed method compared to the methods: wavelet-based, SVD-based features and the corresponding semi-supervised method. Here, for the wavelet-based method, $\psi = 0.02$ and $\xi = 0$; and for SVD-based method, $\psi = 0.005$ and $\xi = 0$. The column of similar image represents the true authentication capability of the judgment of a similar image, which indicates the robustness of the algorithm. The column of tampering image represents the true authentication capability of tampering image, which indicates the discrimination of the algorithm. Higher values mean better robustness and differentiation. Only our approach selected adaptive thresholds, as other approaches choose a fixed threshold that balances robustness and discrimination.

Table 4. Result for the probability of true authentication capability.

Method	Similar Images	Tampered Image
DWT	95.64%	95.81%
Semi-Supervised (DWT)	95.65%	95.78%
OUR (DWT)	96.19%	97.14%
SVD	84.97%	84.92%
Semi-Supervised (SVD)	85.12%	85.08%
OUR (SVD)	86.06%	85.46%

5. Domains of Application

With the aid of sophisticated photoediting software, multimedia content security is becoming increasingly prominent. By using image editing tool, such as Photoshop, the counterfeiters can easily tamper the color attribute to distort the actual meanings of images. Figure 5 shows some real examples for image tamper. These edited images spread over the social network, which not only disturb our daily lives, but also seriously threat our social harmony and stability. If tampered images are extensively used in the official media, scientific discovery, and even forensic evidence, the degree of trustworthiness will undoubtedly be reduced, thus having a serious impact on various aspects of society.

(a) Original image (b) Forged image

Figure 5. The German-language daily tabloid, Blick, forged the flooding water to blood red, and distributed the falsified image to news channels.

Many image hashing algorithms are widely used in image authentication, image copy detection, digital watermarking, image quality assessment and other fields, as shown in Figure 6. Perceptual image hashing aims to be smoothly invariant to small changes in the image (rotation, crop, gamma correction, noise addition, adding a border). This is in contrast to cryptographic hash functions that are designed for non-smoothness and to change entirely if any single bit changes. Our proposed perceptual image hashing algorithm is mainly for image authentication applications. Our technique is suitable for processing large image data, making it a valuable tool for image authentication applications.

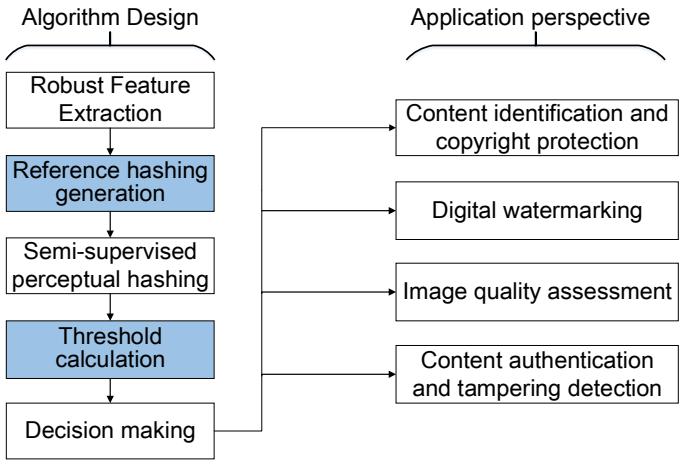


Figure 6. A generic framework of image hashing and an application perspective.

6. Conclusions

In this paper, we have proposed a hashing algorithm based on multi-attack reference generation and adaptive thresholding for image authentication. We effectively exploited simultaneously the supervised content-preserving images and multiple attacks for feature generation and the hashing learning. We specially take into account the pairwise variations among different originally-received image pairs, which makes the threshold more adaptable and the value more reasonable. We performed extensive experiments on two image datasets and compared our results with the state-of-the-art hashing baselines. Experimental results demonstrated that the proposed method yields superior performance. For image hashing-based authentication, a scheme with not only high computational efficiency but also reasonable authentication performance is expected. Compared with other original image-based reference generation, the limitation of the our work is that it is time consuming for cluster operation. In the future work, we will design the co-regularized hashing for multiple features, which is expected to show even better performance.

Author Contributions: Conceptualization, L.D.; methodology, L.D., Z.H. and Y.W.; software, L.D., Z.H. and X.W.; validation, Z.H. and Y.W.; formal analysis, L.D., X.W. and A.T.S.H.; data curation, Z.H. and Y.W.; writing-original draft preparation, L.D., Z.H. and Y.W.; writing-review and editing, X.W., and A.T.S.H.; funding acquisition, L.D. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant number 61602344), and the Science and Technology Development Fund of Tianjin Education Commission for Higher Education, China (grant number 2017KJ091, 2018KJ222).

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this article.

References

1. Kalker, T.; Haitsma, J.; Oostveen, J.C. Issues with DigitalWatermarking and Perceptual Hashing. *Proc. SPIE* **2001**, *4518*, 189–197.
2. Tang, Z.; Chen, L.; Zhang, X.; Zhang, S. Robust Image Hashing with Tensor Decomposition. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 549–560. [CrossRef]
3. Yang, H.; Yin, J.; Yang, Y. Robust Image Hashing Scheme Based on Low-Rank Decomposition and Path Integral LBP. *IEEE Access* **2019**, *7*, 51656–51664. [CrossRef]
4. Tang, Z.; Yu, M.; Yao, H.; Zhang, H.; Yu, C.; Zhang, X. Robust Image Hashing with Singular Values of Quaternion SVD. *Comput. J.* **2020**. [CrossRef]
5. Tang, Z.; Ling, M.; Yao, H.; Qian, Z.; Zhang, X.; Zhang, J.; Xu, S. Robust Image Hashing via Random Gabor Filtering and DWT. *CMC Comput. Mater. Contin.* **2018**, *55*, 331–344.
6. Karsh, R.K.; Saikia, A.; Laskar, R.H. Image Authentication Based on Robust Image Hashing with Geometric Correction. *Multimed. Tools Appl.* **2018**, *77*, 25409–25429. [CrossRef]

7. Gharde, N.D.; Thounaojam, D.M.; Soni, B.; Biswas, S.K. Robust Perceptual Image Hashing Using Fuzzy Color Histogram. *Multimed. Tools Appl.* **2018**, *77*, 30815–30840. [CrossRef]
8. Tang, Z.; Yang, F.; Huang, L.; Zhang, X. Robust image hashing with dominant dct coefficients. *Optik Int. J. Light Electron Opt.* **2014**, *125*, 5102–5107. [CrossRef]
9. Lei, Y.; Wang, Y.-G.; Huang, J. Robust image hash in radon transform domain for authentication. *Signal Process. Image Commun.* **2011**, *26*, 280–288. [CrossRef]
10. Tang, Z.; Dai, Y.; Zhang, X.; Huang, L.; Yang, F. Robust image hashing via colour vector angles and discrete wavelet transform. *IET Image Process.* **2014**, *8*, 142–149. [CrossRef]
11. Ouyang, J.; Coatrieux, G.; Shu, H. Robust hashing for image authentication using quaternion discrete fourier transform and log-polar transform. *Digit. Signal Process.* **2015**, *41*, 98–109. [CrossRef]
12. Yan, C.-P.; Pun, C.-M.; Yuan, X. Quaternion-based image hashing for adaptive tampering localization. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2664–2677. [CrossRef]
13. Yan, C.-P.; Pun, C.-M. Multi-scale difference map fusion for tamper localization using binary ranking hashing. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2144–2158. [CrossRef]
14. Wang, P.; Jiang, A.; Cao, Y.; Gao, Y.; Tan, R.; He, H.; Zhou, M. Robust image hashing based on hybrid approach of scale-invariant feature transform and local binary patterns. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 19–21 November 2018; pp. 1–5.
15. Qin, C.; Hu, Y.; Yao, H.; Duan, X.; Gao, L. Perceptual image hashing based on weber local binary pattern and color angle representation. *IEEE Access* **2019**, *7*, 45460–45471. [CrossRef]
16. Yan, C.-P.; Pun, C.-M.; Yuan, X.-C. Adaptive local feature based multi-scale image hashing for robust tampering detection. In Proceedings of the TENCON 2015—2015 IEEE Region 10 Conference, Macao, China, 1–4 November 2015; pp. 1–4.
17. Yan, C.P.; Pun, C.M.; Yuan, X.C. Multi-scale image hashing using adaptive local feature extraction for robust tampering detection. *Signal Process.* **2016**, *121*, 1–16. [CrossRef]
18. Pun, C.-M.; Yan, C.-P.; Yuan, X. Robust image hashing using progressive feature selection for tampering detection. *Multimed. Tools Appl.* **2017**, *77*, 11609–11633. [CrossRef]
19. Qin, C.; Chen, X.; Luo, X.; Xinpeng, Z.; Sun, X. Perceptual image hashing via dual-cross pattern encoding and salient structure detection. *Inf. Sci.* **2018**, *423*, 284–302. [CrossRef]
20. Monga, V.; Evans, B. Robust perceptual image hashing using feature points. In Proceedings of the 2004 International Conference on Image Processing, Singapore, 24–27 October 2004; pp. 677–680.
21. Qin, C.; Chen, X.; Dong, J.; Zhang, X. Perceptual image hashing with selective sampling for salient structure features. *Displays* **2016**, *45*, 26–37. [CrossRef]
22. Qin, C.; Sun, M.; Chang, C.-C. Perceptual hashing for color images based on hybrid extraction of structural features. *Signal Process.* **2018**, *142*, 194–205. [CrossRef]
23. Anitha, K.; Leveenbose, P. Edge detection based salient region detection for accurate image forgery detection. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2015; pp. 1–4.
24. Kozat, S.S.; Mihcak, K.; Venkatesan, R. Robust perceptual image hashing via matrix invariances. In Proceedings of the 2004 International Conference on Image Processing, Singapore, 24–27 October 2004; pp. 3443–3446.
25. Ghouti, L. Robust perceptual color image hashing using quaternion singular value decomposition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 3794–3798.
26. Abbas, S.Q.; Ahmed, F.; Zivic, N.; Ur-Rehman, O. Perceptual image hashing using svd based noise resistant local binary pattern. In Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, Lisbon, Portugal, 18–20 October 2016; pp. 401–407.
27. Tang, Z.; Ruan, L.; Qin, C.; Zhang, X.; Yu, C. Robust image hashing with embedding vector variance of lle. *Digit. Signal Process.* **2015**, *43*, 17–27. [CrossRef]
28. Sun, R.; Zeng, W. Secure and robust image hashing via compressive sensing. *Multimed. Tools Appl.* **2014**, *70*, 1651–1665. [CrossRef]
29. Liu, H.; Xiao, D.; Xiao, Y.; Zhang, Y. Robust image hashing with tampering recovery capability via low-rank and sparse representation. *Multimed. Tools Appl.* **2016**, *75*, 7681–7696. [CrossRef]

30. Tang, Z.; Zhang, X.; Li, X.; Zhang, S. Robust image hashing with ring partition and invariant vector distance. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 200–214. [CrossRef]
31. Srivastava, M.; Siddiqui, J.; Ali, M.A. Robust image hashing based on statistical features for copy detection. In Proceedings of the IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering, Varanasi, India, 9–11 December 2017; pp. 490–495.
32. Huang, Z.; Liu, S. Robustness and discrimination oriented hashing combining texture and invariant vector distance. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 1389–1397.
33. Zhang, D.; Chen, J.; Shao, B. Perceptual image hashing based on zernike moment and entropy. *Electron. Sci. Technol.* **2015**, *10*, 12.
34. Chen, Y.; Yu, W.; Feng, J. Robust image hashing using invariants of tchebichef moments. *Optik Int. J. Light Electron Opt.* **2014**, *125*, 5582–5587. [CrossRef]
35. Hosny, K.M.; Khedr1, Y.M.; Khedr, W.I.; Mohamed, E.R. Robust image hashing using exact gaussian-hermite moments. *IET Image Process.* **2018**, *12*, 2178–2185. [CrossRef]
36. Lv, X.; Wang, A. Compressed binary image hashes based on semisupervised spectral embedding. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1838–1849. [CrossRef]
37. Bondi, L.; Lameri, S.; Guera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. Tampering detection and localization through clustering of camera-based cnn features. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1855–1864.
38. Yarlagadda, S.K.; Güera, D.; Bestagini, P.; Zhu, F.M.; Tubaro, S.; Delp, E.J. Satellite image forgery detection and localization using gan and one-class classifier. *arXiv* **2018**, arXiv:1802.04881.
39. Du, L.; Chen, Z.; Ke, Y. Image hashing for tamper detection with multi-view embedding and perceptual saliency. *Adv. Multimed.* **2018**, *2018*, 4235268. [CrossRef]
40. Wang, Y.; Zhang, L.; Nie, F.; Li, X.; Chen, Z.; Wang, F. WeGAN: Deep Image Hashing with Weighted Generative Adversarial Networks. *IEEE Trans. Multimed.* **2020**, *22*, 1458–1469. [CrossRef]
41. Wang, Y.; Ward, R.; Wang, Z.J. Coarse-to-Fine Image DeHashing Using Deep Pyramidal Residual Learning. *IEEE Signal Process. Lett.* **2020**, *22*, 1295–1299. [CrossRef]
42. Jiang, C.; Pang, Y. Perceptual image hashing based on a deep convolution neural network for content authentication. *J. Electron. Imaging* **2018**, *27*, 043055. [CrossRef]
43. Peng, Y.; Zhang, J.; Ye, Z. Deep Reinforcement Learning for Image Hashing. *IEEE Trans. Multimed.* **2020**, *22*, 2061–2073. [CrossRef]
44. Du, L.; Wang, Y.; Ho, A.T.S. Multi-attack Reference Hashing Generation for Image Authentication. In Proceedings of the Digital Forensics and Watermarking—18th International Workshop (IWDW 2019), Chengdu, China, 2–4 November 2019; pp. 407–420.
45. Zheng, Z.; Li, L. Binary Multi-View Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1774–1782. [CrossRef] [PubMed]
46. Dong, J.; Wang, W. Casia image tampering detection evaluation database. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2003; pp. 422–426.
47. Korus, P.; Huang, J. Evaluation of random field models in multi-modal unsupervised tampering localization. In Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security, Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6.
48. Korus, P.; Huang, J. Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 809–824. [CrossRef]
49. Venkatesan, R.; Koon, S.M.; Jakubowski, M.H.; Moulin, P. Robust image hashing. In Proceedings of the IEEE International Conference on Image Processing, Thessaloniki, Greece, 7–10 October 2001; pp. 664–666.