JAIDM

Journal of AI and Data Mining
Vol 3, No 1, 2015, 21-30.
doi:10.5829/idosi.JAIDM.2015.03.01.03

Image authentication using LBP-based perceptual image hashing

R. Davarzani[1*], S. Mozaffari[2] and Kh.Yaghmaie[2]

1. Department of Electrical & Computer Engineering, College of Engineering, Shahrood Branch, Islamic Azad University, Shahrood, Iran.
2. Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran.

Received 20 October 2014; Accepted 18 May 2015

**Corresponding author:re.davarzani@gmail.com (R.Davarzani).*

Abstract

Feature extraction is a main step in all perceptual image hashing schemes in which robust features will lead to better results in perceptual robustness. Simplicity, discriminative power, computational efficiency and robustness to illumination changes are counted as distinguished properties of Local Binary Pattern features. In this paper, we investigate the use of local binary patterns for perceptual image hashing. In feature extraction, we propose to use both sign and magnitude information of local differences. So, the algorithm utilizes a combination of gradient-based and LBP-based descriptors for feature extraction. To provide security needs, two secret keys are incorporated in feature extraction and hash generation steps. Performance of the proposed hashing method is evaluated with an important application in perceptual image hashing scheme: image authentication. Experiments are conducted to show that the present method has acceptable robustness against perceptual content-preserving manipulations. Moreover, the proposed method has this capability to localize the tampering area, which is not possible in all hashing schemes.

Keywords: *Center-symmetric Local Binary Patterns, Perceptual Image Hashing, Image Authentication, Tamper Detection.*

1. Introduction

In recent years, we have witnessed the development of multimedia information in many aspects of our daily lives. Multimedia finds its application in various areas including, but not limited to, advertisements, art, entertainment, engineering, medicine, business, scientific research and spatial temporal applications. Many of the advantages of digital multimedia have led to the fast progress on media acquisition tools, powerful hardware, sophisticated editing software and network technologies that provide various media sharing and streaming services. However, digital multimedia data have suffered from illegal access and unauthorized distributions. Professional forgers with advanced technology can alter multimedia data without any trail on forged information. Therefore, it is necessary to create new tools and techniques to discover the authenticity and integrity of digital media [1]. In recent decade, several methods have been extensively studied for intellectual property protection of digital images and image forgeries detection including image watermarking-based schemes [2], digital image forensic-based schemes [3] and perceptual image hashing- based schemes [4, 5].

For security/authentication of multimedia data, perceptual image hashing is introduced based on traditional cryptosystems. Traditional cryptographic hash functions have been used in applications involving data integrity issues and data retrieval [6]. However, it should be noted that the purpose of a cryptographic hash function and a perceptual image hash function are totally different. Hash functions in traditional cryptosystems are very secure, but they are very sensitive, i.e. changing even one bit of the input will change the output considerably. However, digital images should undergo content-preserving manipulations such as compression, enhancement, cropping, and scaling. An image hash function should tolerate such changes and produce hash values similar to the original image hash values with the same visual appearance [7]. On the other hand, the perceptual hashing system should be sensitive to content-changing distortions and reject malicious manipulations and attacks. Perceptual image hash functions generally consist of two steps, as illustrated in figure 1 [8]. The first step extracts a feature vector from the image which is depending on the image content or characteristics itself. In the second step, this feature vector is compressed and quantized into a binary or real number sequence to form the final hash value. Since an image hash also serves as a secure tag of the image, a secret key is incorporated into either feature extraction or hash generation or both to guarantee that the hash is hardly obtained by unauthorized adversaries without the secret key.

This paper is organized as follows. Section 2 and 3 review the background literature of perceptual image hashing methods and center-symmetric local binary patterns, respectively. Section 4 presents the proposed new image hashing scheme. Experimental results are given in section 5. Finally, conclusion is drawn in section 6.

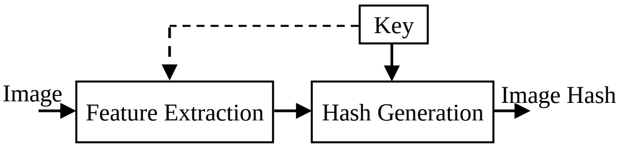


Figure 1. Two main stages of the general image hashing scheme.

2. Related works

Feature extraction is the key step in the image hashing which differentiates perceptual hashing methods. The aim of feature extraction is to present a compact and robust representation of the image content. The extracted features are quantized into a binary or real numbers sequence to form the final hash value.

By reviewing the literature, the existing hash generation algorithms can be roughly classified into three categories based on their feature extraction method: transform based schemes, matrix factorization based schemes, and local feature pattern based schemes.

2.1. Transform based schemes

Many approaches are utilized classic image transforms to extract image features. Various properties of the DCT can be utilized to create perceptual image hash functions.

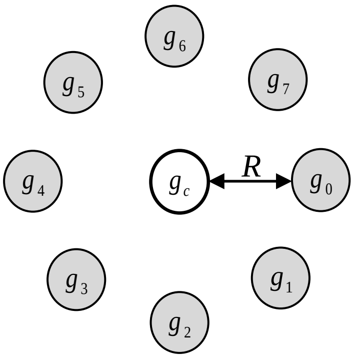
For example, low-frequency DCT coefficients of an image are mostly stable under image manipulations. Fridrich and Goljan [9] proposed a robust hashing algorithm based on the stability of low-frequency DCT coefficients. Another method using scale- and rotation-invariant property of Fourier-Mellin transform (FMT) was proposed in [10]. The authors presented a robust image hashing method using the magnitudes of the Fourier transform coefficients which were randomly weighted and summed. Their method is robust to various content-preserving manipulations such as geometric distortions, filtering operations, and etc. In a more recent development, several methods have been proposed by Radon transform.

To provide robustness against geometric distortions in image hashing, the Radon transform was first used in [11]. An expanded method was proposed in [12]. The method uses Radon transform to divide an image into radial projections and build a RAdial Variance (RAV) vector of image pixels. Then, the first 40 DCT coefficients of the RAV vectors are converted into the robust image hash called RASH. The method is computationally simple and is resilient to scaling and rotation but its discriminative capability needs to be strengthened. Venkatesan et al. [13] proposed a perceptual image hashing technique based on quantized statistics of randomized rectangles in the discrete wavelet domain (DWT). In their method, averages or variances of the random blocks in wavelet image are computed and then quantized with randomized rounding to form a secure binary hash. This method is sensitive to contrast adjustment and gamma correction and robust against a limited range of geometric attacks.

2.2. Matrix factorization based schemes

In the second type of image hashing schemes, the advantage of matrix factorization or decomposition are used to extract the image features. In this category singular value decomposition (SVD) and non negative matrix factorization (NMF) are the most popular decomposition tools for image matrixes. Kozat et al. [14] proposed a hashing algorithm in two steps using SVD.

The method is robust against some small variations in rotation and scaling. In another work in [15], Non-negative Matrix Factorization (NMF) is used as a dimensionality reduction technique to generate image hashing. Although, the method is resilient to a large class of perceptually insignificant attacks but it is vulnerable to brightness changes and large hashing methods using dimension reduction techniques, a new robust and secure image hash algorithm was also proposed based on Fast Johnson-Lindenstrauss transform (FJLT) [16].



$$CSLBP_{8,R,T}(g_c) = 2^0 \times s(g_0 - g_4) + 2^1 \times s(g_1 - g_5) + 2^2 \times s(g_2 - g_6) + 2^3 \times s(g_3 - g_7)$$

Figure 2. Circularly symmetric neighborhoods and CSLBP feature for radius R and P=8 neighborhood pixels.

2.3. Local feature pattern based schemes

Finally, the methods like [17], [18] and [19] can be included in the third group of image hashing algorithms called local feature patterns-based hashing schemes. In [18] and [19] SIFT keypoints are used as most significant and robust local features for image hashing.

In this work we propose a novel and robust perceptual image hashing method based on Center-Symmetric Local Binary Pattern (CSLBP) features. CSLBP features are extracted from each non-overlapping block of the original gray-scale image. For each block, the final hash code is obtained by inner product of its CSLBP feature vector and a pseudorandom weight vector. To provide security needs, two secret keys are incorporated in feature extraction and hash generation steps. Many experiments are conducted and the results show that our algorithm can reach a good balance between robustness and discrimination and outperforms some well-known algorithms. High robustness against luminance changes, applicability in image tampering detection, acceptable hash length and running time can be also counted as the advantage of our proposed hashing method.

3. Center-symmetric local binary patterns (CSLBP)

Among the feature descriptors, Local Binary Patterns (LBP) is one of the most famous and powerful ones. The idea of LBP is originally proposed by Ojala et al. [20, 21] for texture classification. Then, it has gained increasing attention in many image analyses applications due to its low computational complexity, invariance to monotonic gray-scale changes and texture description ability [20]. LBP has been utilized in various image analyses applications such as dynamic texture recognition [22], face recognition forgery [25], image region descriptors [26] and so on.

In the original LBP, signed gray level differences of each pixel with its neighboring pixels are described as a binary form. However, the LBP operator produces rather long histograms and it is therefore difficult to use in the context of a region descriptor. Furthermore, the original LBP feature is not robust on flat images. To address the problems, center-symmetric local binary patterns, CSLBP, as a modified version of LBP was proposed in [26].

Let $I(x,y)$ be a gray level image and g_c indicates the gray level of an arbitrary pixel positioned at (x_c, y_c), i.e. $g_c = I(x_c, y_c)$ .Gray values of P equally spaced circular neighborhood pixels on a circle of radius $R(R>0)$, around g_c are shown by $g_p, p = 0,1,...,P$ -1, (See Figure 2). The CSLBP form shown by $CS\_LBP_{P,R,T}(x_c, y_c)$ is obtained as follows:

$$g_p = I(x_p, y_p), \quad p = 0,...,P-1 \tag{1}$$

$$x_p = x_c + R\cos(2\pi p/P)$$

$$y_p = y_c - R\sin(2\pi p/P)$$

$$CSLBP_{P,R,T}(x_c, y_c) = \sum_{p=0}^{P/2-1} s(g_p - g_{p+(P/2)})2^p,$$

$$s(x) = \begin{cases} 1, & \text{x} > T \\ 0, & \text{otherwise} \end{cases}$$

where, T is a small value used to threshold the gray-level difference to increase the robustness of the CSLBP feature on flat image regions. CSLBP captures better gradient information than the basic LBP, because instead of comparing the gray-level of each pixel with the center pixel, gray-level differences between center-symmetric pairs of opposite pixels in a neighborhood are compared.

4. Proposed algorithm for perceptual image hashing

In this section, we propose image hashes generation based on block feature extraction using center-symmetric local binary patterns (CSLBP).

As shown in figure 3, the proposed technique consists of three steps: pre-processing, feature extraction, and hash generation. To guarantee the security requirements, two secret keys, *K1* and *K2*, are also incorporated in feature extraction and hash generation steps. The following section describes details of our hashing algorithms.

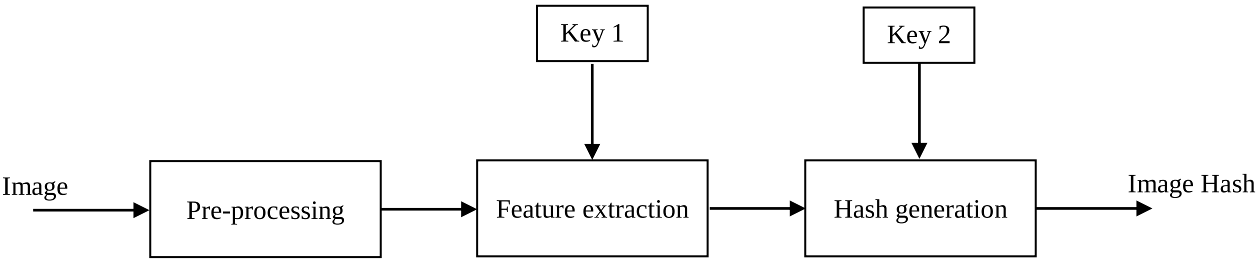


Figure 3. Block diagram of the proposed hashing algorithms.

4.1. Pre-processing

In this work, we are dealing with gray-level images, so RGB images are first converted to grayscale images using standard color space conversion.

Since real images may have different size, to ensure that the final generated hash has a fixed length, all input images are first rescaled into a standard resolution of $M \times N$ by bi-linear interpolation. Then, the resized input image is divided into non-overlapping blocks of $B \times B$ pixels and feature extraction is applied to each image block. We need to keep a trade-off between the hash length, the discriminative capability and perceptual robustness in choosing the size of blocks. This is because a large block size means few features which will inevitably reduce discriminative capability. When the size of block is decreased, discrimination can be improved but perceptual robustness is easily affected by minor modification. In addition, a smaller block size will increase the hash size. In experiments, we find that $B = 32$ is an acceptable moderate size for 256×256 images.

In this scheme, before feature extraction, we first filter each block with an edge-preserving adaptive low-pass filter. The adaptive filter is more selective than a comparable linear filter, because of preserving edges and other high-frequency parts of an image. For this purpose, we use a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. Our experiments have shown that this filtering has considerably improvements on robustness of image hash. The consecutive operations in the pre-processing step are drawn in figure 4.

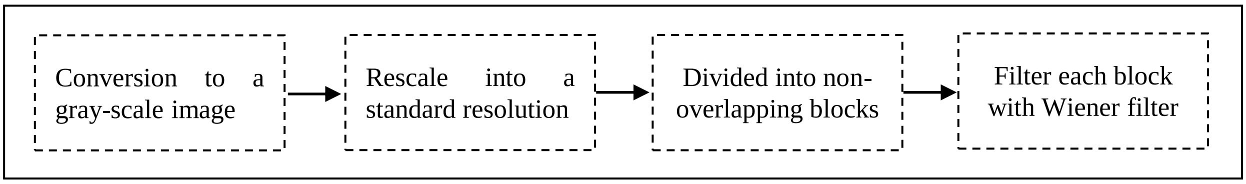

Conversion to a gray-scale image
Rescale into a standard resolution
Divided into non-overlapping blocks
Filter each block with Wiener filter

Figure 4. Pre-processing steps in our algorithm.

4.2. Feature extraction

In CSLBP-based image hashing, CSLBP features are extracted to represent the main content of the image compactly. Then, in hash generation step, the extracted features are converted into a real number sequence to form the final hash value. The details of these feature extraction method is presented as follows.

Given a central pixel g_c and its P equally spaced circular neighborhood pixels $g_p, p = 0,1,...,P\text{-}1$, we can simply calculate gray-level differences between center-symmetric pairs of opposite pixels in a neighborhood as (2):

$$d_{p,q} = g_p - g_q, \tag{2}$$
$$q = p + (P/2), p = 0,1,...,(P/2-1)$$

$d_{p,q}$ can be further decomposed into two components:

$$d_{p,q} = s_{p,q} \times m_{p,q} \qquad (3)$$

$$s_{p,q} = \text{sign}(g_p - g_q),\ \ \text{sign}(x) = \begin{cases} 1, & x \ge 0 \\ -1, & x < 0 \end{cases}$$

$$m_{p,q} = \left| g_p - g_q \right|$$

Where $s_{p,q}$ and $m_{p,q}$ are the sign and the magnitude of $d_{p,q}$, respectively. The original CSLBP operator discards the magnitude information of the difference between the center-symmetric pairs of pixels and uses only the sign information (see (1)). In other words, for a given $M \times N$ image, after identifying the CSLBP pattern of each pixel (i, j), the normalized histogram of CSLBP codes is computed over the image and it is used as a feature vector, (4):

$$H(b) = (\frac{1}{M \times N})\sum_{i=1}^{M}\sum_{j=1}^{N} f\,(CSLBP_{P,R,T}\,(i,j), b), \qquad (4)$$

$$b \in [0, B]$$

$$f\,(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases}$$

where, *B* is the maximal CSLBP pattern value.

In a gradient-based descriptor, image gradients are used to obtain magnitude and orientation for each image pixel. Since, definition of magnitude in CSLBP operator, $m_{p,q}$, is similar to image gradient, it contains useful information for image description [27].

In our proposed CSLBP-based image hashing, the motivation behind proposed feature descriptor is an efficient combination of CSLBP-based descriptor and gradient-based descriptor using sign and magnitude information of local differences. We extract two features for each pixel of the image by using the CSLBP operator.

The sign feature is the same as the original CSLBP code defined by (1) and the magnitude feature vector corresponds to gradient information which is achieved by the magnitude of local differences.

The CSLBP operator has three parameters: radius *R*, number of neighboring pixels *P*, and threshold on the gray level difference *T*. In our experiments best results are achieved by *R*=1, *P*=8, and *T*=0.01. Using 8 neighboring samples, for each pixel (i, j), the sign feature (CSLBP code) can get 16 decimal numbers from 0 to 15, and the magnitude feature vector, *MV*, is defined by (5):

$$CS\_LBP_{8,1,0.01}(i,j) = \sum_{p=0}^{3} s(g_p - g_{p+4})2^p \qquad (5)$$

$$MV\,(i,j) = [m_{0,4}, m_{1,5}, m_{2,6}, m_{3,7}]$$

where, $m_{p,q}$, $q = p + 4$, $p = 0,1,2,3$ is defined by (2) and (3).

Four histograms are built considering four components of magnitude vector. Each pixel of the image with a given CSLBP code is assigned to a bin in the histogram according to its magnitude. In other words, the magnitude feature, *MV*, is used as an adaptive weight in histogram calculation of CSLBP codes, (6).

$$H_p(b) = \sum_{i=1}^{B}\sum_{j=1}^{B} m_{p,q}(i,j) \times f\,(CSLBP(i,j), b), \qquad (6)$$

$b \in [0,15]$, $p = 0,1,2,3$ and $q = p + 4$.

Finally, the obtained histograms are joined together to create the feature vector of an image block, *FV*, which is used for hash generation, (7). Each *FV* has 64 elements and to enhance the security of the scheme, all the elements in each *FV* are randomly scrambled according to a secret key *K*1.

$$FV = [H_0, H_1, H_2, H_3] \qquad \cdots$$

where, $\langle U, V \rangle$ indicates the inner product of two vectors *U* and *V*.

4.3. Hash generation

In the previous step, a feature vector was constructed for each non-overlapping block of input image. We generate pseudorandom weights $\omega = \{\alpha_i\}, i = 1,\ldots,64$ from the normal distribution $N\,(u, \sigma^2)$ using a secret key, *K*2. ω is a random vector with 64 dimensions, with the same size of *FV*. Let $H = \{h_b\}, b = 1,\ldots,N$ be the hash vector of input image where *N* is the total number of non-overlapping image blocks; we define *FV*$_b$ as feature vector of *b*th block and its corresponding h_b component by (8):

$$h_b = \langle FV_b, \omega \rangle \qquad \cdots$$

where, $\langle U, V \rangle$ indicates the inner product of two vectors *U* and *V*.

5. Experiments

In this section, we provide a comprehensive evaluation of the proposed algorithm in image authentication [19]. Furthermore, some experiments are also included to analyze the performances of the proposed hashing scheme with respect to forged region detection.

5.1. Evaluation of image authentication

Image authentication experiment is designed to measure the sensitivity of our method to distinguish malicious attacks from content-preserving distortions. Figure 5 illustrates this usage mode for perceptual image hashing. In image authentication, the hash of an original image, H_{org}, is available and called the reference hash. The hash of a test image, H_{test}, is extracted using the same perceptual image hashing algorithm. Then, these two hashes are compared. Now, the image in question is declared to be authentic if $d\left(H_{test}, H_{org}\right) < T$, where $d(.)$ is a distance measure and T is a predefined threshold.

In our method, since hash values are approximately linearly changed, the correlation coefficient is used as a distance measure.

We considered the problem of image authentication as a hypothesis testing problem with two hypotheses: (H_0: *Image is authentic*) and (H_1: *Image is not authentic*). Each test image is classified into one of the hypothesis states. We use the receiver operating characteristics (ROC) curve to examine the discriminative capabilities of various hashing schemes in image authentication. True positive rate (TPR) and false positive rate (FPR) are two axes of ROC curve, which are defined by (9) and (10), respectively.

$$TPR(T) = \frac{\text{Number of true images detected as authentic images}}{\text{Total number of authentic images}} \qquad \cdots$$

$$FPR(T) = \frac{\text{Number of forged images detected as authentic images}}{\text{Total number of forged images}} \qquad \cdots\cdot$$

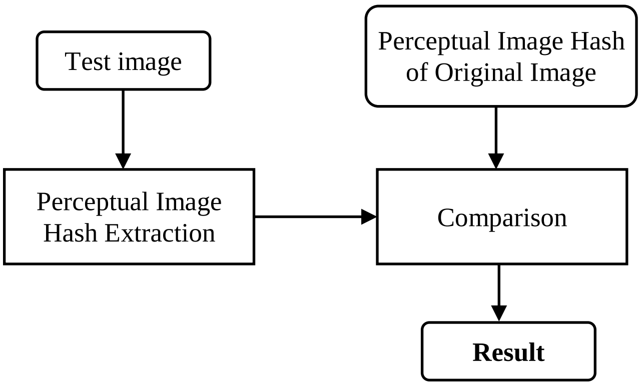

Test image
Perceptual Image Hash of Original Image
Perceptual Image Hash Extraction
Comparison
Result

Figure 5. Image authentication framework.

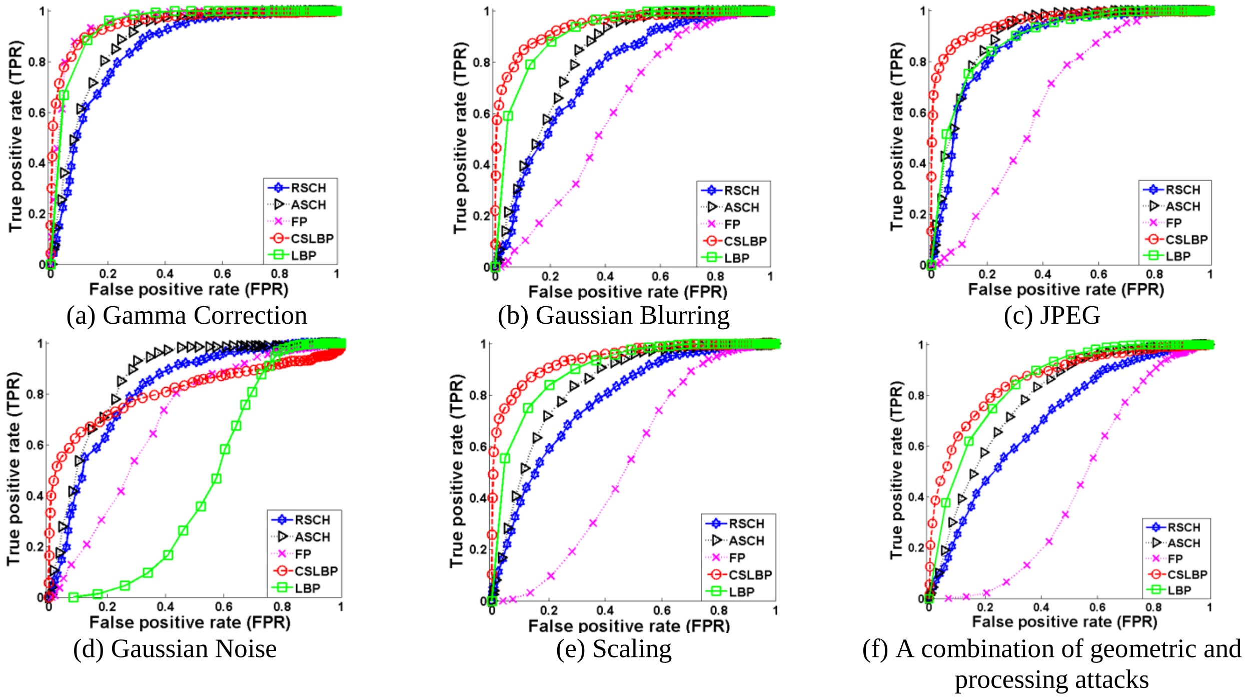
Based on a given threshold (T), TPR gives us an estimate of the probability of true detection and FPR obtains the percentage of images that are falsely classified as original image.

To implement authentication experiment, we construct three databases: an *original images database*, a *similar images database* and a *forged images database*. The original images database is constructed by a collection of 1000 color images from two databases: The Corel database [28] and Columbia photographic images and photorealistic computer graphics dataset [29]. All the color images are resized to 256×256 and converted as gray-scale images in the experiments. Similar images are obtained by applying several content-preserving operations on each original image. The operations are listed in table 1. Our forged image database is constructed by splicing image forgery. Image splicing is a simple form of photomontage technology where is defined by simple combining image fragments from two or more different images without further post-processing. A total of 1000 tampered images are created with perform splicing forgery on each original image. In each forged image the pasted area is 10% of the host image.

Figure 6 compares the ROC curves of the proposed method and those of [30] and two methods in [19]. Refer to (9) and (10), it is clear that TPR and FPR indicate robustness and discrimination, respectively. It can be observed from figure 6 (f) that the proposed method has shown stronger ability than the other four methods to distinguish content-preserving distortions from malicious attack. For example with the same probability of false detection in FPR = 0.2, CSLBP achieves higher probability of correct detection (TPR = 0.77) than other hashing methods. In the same FPR, the TPR for RSCH [19], ASCH [19] and FP [30] hashing methods are 0.46, 0.58 and 0.022 respectively.

Table1. Content-preserving manipulations with some details in parameters description and setting.

Manipulation	Parameter description	Parameter setting
Gaussian noise	Mean (*m*), Variance (*v*)	$m = 0,\ v = 0.0005$
Gaussian blurring	Standard deviation (σ) , window size (F_s)	$F_s = 3, \sigma = 0.5$
Gamma correction	Gamma (γ)	$\gamma = 1.1$
Scaling	Scaling factor (s)	$s = 1.5$
JPEG Compression	Quality factor (*Q*)	$Q = 70$
A combination of attacks		$(\gamma = 1.1)$, $(Q = 70)$, $(\sigma = 0.5, F_s = 3)$ and $(s = 1.5)$.



(a) Gamma Correction (b) Gaussian Blurring (c) JPEG

(d) Gaussian Noise (e) Scaling (f) A combination of geometric and processing attacks

Figure 6. ROC curve comparisons between our hashing and other methods in image authentication test.

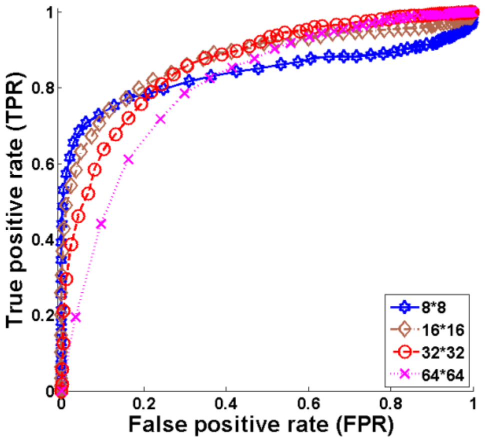


Figure 7. ROC curve comparisons between different block sizes.

5.2. Effect of block size

To show effect of block size on hash performances of our methods, we change the block size and conduct the experiment under a combination of geometric and processing attacks, (in the last row of Table 1). The used block sizes are: 8 × 8, 16 × 16, 32 × 32 and 64 × 64. The ROC graphs are shown in figure 7. We can observe that all block sizes have similar ROC curves. However, based on the hash length and the area of under ROC curves, block size 32 × 32 is an acceptable compromise for 256 × 256 images.

5.3. Tampering localization

A forger can easily modify local regions of an image and alter its original content by available photo-editing software. In digital image forensics, many techniques have been proposed to address the problem of forged region detection [3, 31]. Block-based matching is one of the main methods in forged region localization. In the method, first, the image is divided into overlapping or non-overlapping blocks. Then, the perceptual hash function generates a hash value for each block of the original image. During forensic analysis, hashing codes are extracted from the corresponding blocks of the suspect image and a block-wise comparison can reveal potential tampered regions. For tampering localization, the choice of block size controls the trade-off between hash length and detection performance. Larger block size gives a smaller hash length but can introduce higher false detection than a smaller block size. An illustration of tampering localization functionality of the proposed method is provided by the following experiment. We used a database of 100 image pairs, which includes the original image and tampered copy. Tampered images are generated by splicing technique where some regions of the original image are replaced with foreign blocks. We assess the accuracy of tampering localization by ROC analysis. In the results, the receiver operating curve is a plot of the probability of true positive rate versus the false positive rate as the system threshold is varied. The two probabilities are defined as follows:

$$TPR(T) = \frac{\text{Number of tampered blocks detected as tampered}}{\text{Total number of tampered blocks}} \qquad \cdots\cdot$$

$$FPR(T) = \frac{\text{Number of genuine blocks detected as tampered}}{\text{Total number of genuine blocks}} \qquad \cdots\cdot$$

where, T is the variant threshold parameter.

The result of ROC analysis is presented in figure 8, in which the true positive rates and false positive rates are averaged based on the results from 100 image pairs. The figure compares the sensitivity of forged region detection in our method and the method in [18]. Roy et al. uses quantized edge direction histogram features to localize tempered blocks. We can observe from figure 8 that the proposed method attain better accuracy than [18]. Because our method for hash generation utilize combine LBP-based descriptors and gradient-based descriptors. It is worth noting that FP [30] is not applicable for tampering detection. In this sense, the proposed hashing scheme is more generally applicable.

An example of forged region detection, using CSLBP-based hashing method, is also illustrated in figure 9. From left to right, columns 1 to 3 are original images, tampered images and detection results, respectively.

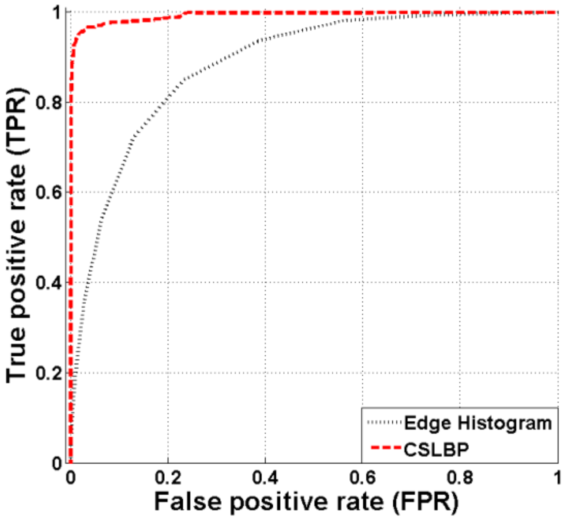


Figure 8. The ROC curve for tampering localization.

5.4. Hash length and CPU running time

Clearly, hash length in the methods based on the block feature extraction is proportional to the number of image blocks. In the local feature point-based schemes, the number of keypoints determines the hash length. According to the parameters used in the experiment, the hash length of different methods is listed in table 2.

The hash length of our method is 64 decimal digits. Table 2 also compares the average time of producing each image hash in different methods. In implementation we used a Sony Vaio laptop, Intel Core 2 Duo Processor P8800 (2.66 GHz), memory 4 GB and software of MATLAB 7.7.0.

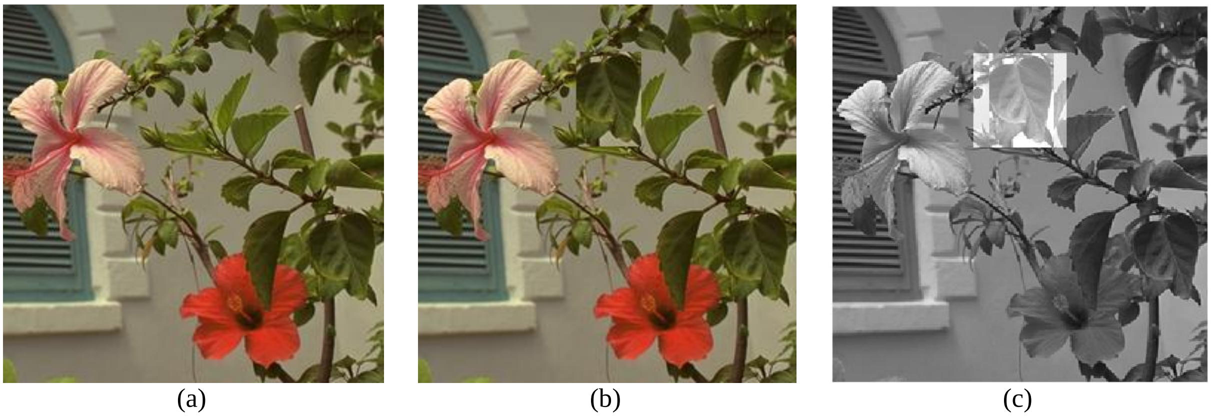


Figure 9. Visualized forged detection results using CSLBP-based hashing method. From the left, columns 1 to 3 are original images, tampered images and detection results, respectively.

An example of forged region detection, using CSLBP-based hashing method, is also illustrated in figure 9. From left to right, columns 1 to 3 are original images, tampered images and detection results, respectively.

5.4. Hash length and CPU running time

Clearly, hash length in the methods based on the block feature extraction is proportional to the number of image blocks. In the local feature point-based schemes, the number of keypoints determines the hash length. According to the parameters used in the experiment, the hash length of different methods is listed in table 2. The hash length of our method is 64 decimal digits. Table 2 also compares the average time of producing each image hash in different methods. In implementation we used a Sony Vaio laptop, Intel Core 2 Duo Processor P8800 (2.66 GHz), memory 4 GB and software of MATLAB 7.7.0. The processing time of hash generation is computed through the average time on 100 images of size 256×256. Since the algorithms in [19] spent a computation time on the robust local keypoint detection, the average time in ASCH and RSCH is higher than the other methods. Our fast hashing method can be attributed to the low computational complexity of LBP features.

Table 2. Time and hash length comparisons among different algorithms

Method	Hash length	Average time (sec)
SCH	20 decimal digits	3.55
CSLBP	64 decimal digits	0.1

6. Conclusion

In this paper we proposed a new perceptual image hashing method based on local binary patterns (LBP). In this algorithm a simple and efficient version of LBP feature, called center-symmetric LBP (CSLBP), was used for feature extraction. The original CSLBP descriptor uses only the sign information of local differences. In this paper, however, both sign and magnitude information are utilized for image hashing to make benefit of gradient-based and LBP-based descriptor simultaneously. To increase security of our hashing method, two secret keys were used in feature extraction and hash generation steps. To evaluate our proposed method, an experiment was conducted for image authentication. The results demonstrated that our proposed scheme could distinguish legal distortions from malicious manipulations. Finally, applicability in image tampering detection, acceptable hash length and running time can be also counted as the advantage of our proposed hashing method.

References

[1] Haouzia, A. & Noumeir, R. (2008). Methods for image authentication: a survey, Multimedia Tools and Applications, vol. 39, no. 1, pp. 1-46.

[2] Cox, I., M. Fridrich, J. & Kalker, T. (2008). Digital Watermarking and Steganography: Morgan Kaufmann Publishers Inc.

[3] Birajdar, G. K. & Mankar, V. H. (2013). Digital image forgery detection using passive techniques: A survey, Digital Investigation, vol. 10, no. 3, pp. 226-245.

[4] Shuo-zhong W. & Xin-peng, Z. (2007). Recent development of perceptual image hashing, Journal of Shanghai University (English Edition), vol. 11, no. 4, pp. 323-331.

[5] Weng, L. (2012). Perceptual Multimedia Hashing (Ph.D. thesis), Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Heverlee (Belgium).

[6] Rivest, R. (1992). The MD5 Message-Digest Algorithm: RFC Editor.

[7] Wu, M., Mao, Y. & Swaminathan, A. (2007). A Signal Processing and Randomization Perspective of

Robust and Secure Image Hashing, 14th IEEE Workshop on Statistical Signal Processing, SSP '07., pp. 166-170.

[8] Monga, V. (2005). Perceptually based methods for robust image hashing (Ph.D. thesis), in Electrical Engineering, Electrical and Computer Engineering, The University of Texas at Austin.

[9] Fridrich J. & Goljan, M. (2000). Robust hash functions for digital watermarking, International Conference on Information Technology: Coding and Computing, 2000, pp. 178-183.

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

[11] Lefbvre, F., Macq, B. & Legat, J. D. (2002). RASH: RAdon Soft Hash algorithm, 11th European IEEE Signal Processing Conference, 2002 , pp. 1-4.

[12] De Roover, C., De Vleeschouwer, C., Lefebvre, F. & Macq, B. (2005). Robust image hashing based on radial variance of pixels, IEEE International Conference on Image Processing, ICIP, pp. 77-80.

[13] Venkatesan, R., Koon, S., Jakubowski, H. & Moulin, P. (2000). Robust image hashing, International Conference on Image Processing, 2000, pp. 664-666.

[14] Kozat, S., Venkatesan, R. & Mihcak, M. K. (2004). Robust perceptual image hashing via matrix invariants, International Conference on Image Processing, ICIP 2004, pp. 3443-3446.

[15] Monga, V. & M. K. Michkak, (2007). Robust and Secure Image Hashing via Non-Negative Matrix Factorizations, IEEE Transactions on Information Forensics and Security, , vol. 2, no. 3, pp. 376-390.

[16] Xudong, L. & Wang, J. (2008). Fast Johnson-Lindenstrauss Transform for robust and secure image hashing, 10th IEEE Workshop on Multimedia Signal Processing, pp. 725-729.

[17] Monga, V. & Evans, B. L. (2006). Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs, IEEE Transactions on Image Processing, , vol. 15, no. 11, pp. 3452-3465.

[18] Roy, S. & Sun, Q. (2007). Robust Hash for Detecting and Localizing Image Tampering, IEEE International Conference on Image Processing, ICIP 2007, pp. VI - 117-VI - 120.

[19] Xudong, L. & Wang, Z. J. (2012). Perceptual Image Hashing Based on Shape Contexts and Local Feature Points, IEEE Transactions on Information Forensics and Security, , vol. 7, no. 3, pp. 1081-1093.

[20] Ojala, T., Pietikäinen, M. & Harwood, D. (1996), A comparative study of texture measures with classification based on featured distributions, Pattern Recognition, vol. 29, no. 1, pp. 51-59.

[21] Ojala, T., Pietikäinen, M. & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987.

[22] Zhao, G. & Pietikäinen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 915-928.

[23] Ahonen, T., Hadid, A. & Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, , vol. 28, no. 12, pp. 2037-2041, Dec 2006.

[24] Heikkila, M. & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 657-662.

[25] Davarzani, R., Yaghmaie, K. Mozaffari, S. & Tapak, M. (2013). Copy-move forgery detection using multiresolution local binary patterns, Forensic Science International, vol. 231, no. 1-3, pp. 61-72.

[26] Heikkilä, M., Pietikäinen, M. & Schmid, C. (2009). Description of interest regions with local binary patterns, Pattern Recognition, vol. 42, no. 3, pp. 425-436.

[27] Guo, Z. & Zhang, D. (2010). A Completed Modeling of Local Binary Pattern Operator for Texture Classification, IEEE Transactions on Image Processing, vol. 19, no. 6, pp. 1657-1663, 2010.

[28] Corel, test set. [Online]. http://wang.ist.psu.edu/~jwang/test1.tar , (2001).

[29] Ng, T., Chang, S., Hsu, Y. & Pepeljugoski, M. (2005). Columbia Photographic Images and Photorealistic Computer Graphics Dataset.

[30] Monga, V., Vats, D. & Evans, B. L. (2005). Image Authentication Under Geometric Attacks Via Structure Matching, IEEE International Conference on Multimedia and Expo, ICME 2005, pp. 229-232.

[31] Stamm, M. C., Wu, M., & Liu, K. J. R., (2013). Information Forensics: An Overview of the First Decade, IEEE Access, vol. 1, no. 1, pp. 167-200.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

تصدیق تصویر با استفاده از درهم‌سازی ادراکی تصویر برپایه الگوهای باینری محلی

رضا داورزنی[۱*]، سعید مظفری[۲]، و خشایار یغمائی[۲]

[۱]دانشکده مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی واحد شاهرود، شاهرود، ایران.

[۲]دانشکده برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران.

ارسال ۲۰۱۴/۱۰/۲۰؛ پذیرش۲۰۱۵/۰۵/۱۸

چکیده:

استخراج ویژگی بخش مهم و اساسی در بسیاری از الگوریتم‌های درهم‌سازی ادراکی تصویر محسوب می‌شود. انتخاب ویژگی‌های مقاوم‌تر موجب افزایش مقاومت کدهای‌درهمش استخراجی از تصویر می‌گردد. سادگی، قدرت تمایز بالا، بارمحاسباتی پایین و مقاومت در برابر تغییرات روشنایی همگی جزء مزایای روش استخراج ویژگی الگوهای باینری محلی است. در این مقاله استفاده از ویژگی الگوهای باینری محلی در درهم‌سازی ادراکی تصاویر مورد بحث و بررسی قرار گرفته است. در استخراج ویژگی از هر دو جنبه اطلاعات دامنه و علامت تفاضلات محلی پیسکل‌های تصویر استفاده شده است. به منظور ایجاد کدهای درهمش امن، دو کلید امنیتی نیز در روند استخراج کدهای درهم‌سازی مورد استفاده قرار گرفته است. کارایی الگوریتم درهم‌سازی ارائه شده در زمینه تصدیق تصویر و تشخیص ناحیه جعل بررسی شده است. نتایج آزمایش‌ها به همراه مقایسات انجام شده قابلیت‌ها و نقاط ضعف الگوریتم را به خوبی تشریح می‌نماید.

کلمات کلیدی:الگوهای باینری متقارن متحدالمرکز، درهم‌سازی ادراکی تصویر، تصدیق تصویر، تشخیص ناحیه جعلی.