Image Hashing Algorithm for Copy Detection: Postprint

Authors: Shen Qi, Zhao Yan

Date: 2018-05-18T00:00:00+00:00

Abstract

To accurately and rapidly identify copied images, an image hashing algorithm based on CS-LBP (centrally symmetric local binary pattern) texture and bit-plane statistics is proposed. The algorithm first performs preprocessing; then employs three-level wavelet decomposition to obtain approximation images and high-frequency information, applies Ring partition to the second- and third-level approximation images to extract statistical features from each ring, and performs bit-plane decomposition on the horizontal and vertical components of the second- and third-level high-frequency information to extract statistical features; finally, all low-frequency and high-frequency features are concatenated to generate the image hash sequence. Experimental results demonstrate that the proposed algorithm' s classification performance outperforms some existing hashing algorithms, exhibiting excellent accuracy in copy detection applications.

Full Text

Preamble

Image Hash Algorithm for Copy Detection

Shen Qi, Zhao Yan†

(College of Electronics & Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: To accurately and rapidly identify copied images, this paper proposes an image hashing algorithm based on CS-LBP (centrally symmetric local binary pattern) texture and bit image statistics. The algorithm first preprocesses the image, then uses three-level wavelet decomposition to obtain approximation images and high-frequency information. The second and third-level approximation images are segmented using Ring segmentation to extract statistical features from each ring. The horizontal and vertical components of the

second and third-level high-frequency information undergo bit image decomposition to extract statistical features. Finally, all low-frequency and high-frequency features are combined to generate the image hash sequence. Experimental results demonstrate that the proposed algorithm achieves superior classification performance compared to existing hashing algorithms and exhibits excellent accuracy in copy detection applications.

Keywords: CS-LBP; Ring segmentation; bit image; image copy detection

0 Introduction

With the rapid development of network technology, copied images are increasing at a geometric rate. The emergence of editing software such as Meitu Xiuxiu, Baidu Motu, and PhotoMagic enables users to edit images arbitrarily without professional expertise, making image copy detection increasingly difficult. The concept of image hashing, due to its characteristics, offers significant advantages for copy detection applications. Image hashing refers to representing an image as a short sequence through an algorithm. Its properties include robustness, distinctiveness, and security. Robustness means that the hash sequences of an original image and its processed versions should be similar; distinctiveness means that two different images should have different hash sequences; security means that hashes obtained with incorrect keys should be significantly different from those obtained with the correct key.

The formal concept of image copy detection was proposed in 2003 when Kim [1] introduced an image copy detection algorithm based on image blocking and Discrete Cosine Transform (DCT). This method uses DCT coefficients to construct hashes but involves matrix blocking, making it non-robust to rotation. Ling et al. [2] proposed using multi-scale SIFT (scale-invariant feature transform) feature descriptors to construct hashes. This method extracts SIFT feature descriptors with different radii from certain regions, effectively improving the discriminative capability of local areas. Wu et al. [3] proposed a targeted copy image detection algorithm that utilizes the rotation invariance of the Shi-Tomasi corner detection operator to extract rotation-insensitive corners, then calculates feature descriptors within circular regions centered at these corners to construct hashes. This method demonstrates good robustness to cropping and rotation but poor robustness to Gaussian filtering. Tang et al. [4] constructed hashes using DCT coefficients by dividing the image into blocks, performing DCT on each block, and then using DCT coefficients to build invariant features. While this algorithm offers significant computational speed advantages, it is not robust to rotation and its classification performance needs improvement. Tang et al. [5] proposed a hashing algorithm based on PCA (principal component analysis) feature distance. This method divides the image into blocks, unfolds each block column-wise to form a secondary image, and then applies PCA to obtain the hash sequence. Although this algorithm achieves good classification performance, its efficiency is low. Li et al. [6] first used Gabor transforms to construct a structural map of the image, then converted the structural map to

polar coordinates, performed sub-block division and normalization to achieve rotation robustness, and finally performed weighted summation and quantization on structural sub-images to generate the hash sequence. This algorithm achieves good robustness and distinctiveness while ensuring compactness. Srivastava et al. [7] proposed a hashing algorithm based on statistical features, which applies Radon transforms in different directions to the image, performs DCT on each column of Radon coefficients, and finally extracts statistical features (mean, standard deviation, kurtosis, and skewness) from the AC coefficients of each column to obtain the hash sequence. This algorithm demonstrates good robustness to conventional processing but its efficiency needs improvement.

To achieve good classification performance for image hashing in copy detection applications while maintaining high efficiency, this paper utilizes the strong descriptive capability and fast computation of CS-LBP texture to extract statistical features from wavelet decomposition approximation images combined with Ring segmentation. Additionally, leveraging the good noise resistance of bit images, we apply bit image decomposition to high-frequency information from wavelet decomposition to extract statistical features, thereby improving the algorithm' s robustness to noise. The features obtained from approximation images and high-frequency information are then combined to form an intermediate hash, which is finally scrambled to produce the final hash. Experimental results show that the proposed algorithm achieves excellent classification performance.

1 Hash Algorithm Framework

1.1 Hash Algorithm Framework

The proposed hash algorithm consists of four main components: preprocessing, wavelet decomposition, CS-LBP statistical features, and bit image statistical features. The detailed process is illustrated in [Figure 1: see original paper].

1.2 Image Preprocessing

First, images of different sizes are normalized to M×M using bilinear interpolation. Normalizing images to the same size improves the algorithm' s robustness to scaling. Then, Gaussian low-pass filtering is applied to enhance robustness to noise. Since the extracted features are based on texture and bit image statistics, color images are converted to grayscale images.

1.3 Wavelet Decomposition

Three-level wavelet decomposition [8] is applied to the preprocessed image. Since the selected LBP operator has a fixed radius and only covers a fixed region, it cannot capture texture features at different scales. Wavelet decomposition addresses this limitation by providing multi-scale texture information while compressing certain information. In wavelet decomposition, the choice of basis func-

tion affects the algorithm' s robustness and distinctiveness. This algorithm selects the Haar wavelet because it ensures fast computation while preserving most of the image information [9].

1.4 Texture Extraction Based on CS-LBP Operator

The CS-LBP operator [10] defines a new encoding scheme based on the fundamental LBP operator. CS-LBP offers significant computational speed advantages over the LBP operator. The encoding rules for CS-LBP are shown in equations (1) and (2):

$$CS\text{-}LBP_{R,N}(x_c, y_c) = \sum_{i=0}^{(N/2)-1} s(g_i - g_{i+(N/2)})2^i, \quad s(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where R represents the circle radius and N represents the number of pixel points on the circumference.

CS-LBP texture images at the second and third levels are obtained by applying the CS-LBP operator to the approximation images from the three-level wavelet decomposition.

1.5 Feature Extraction Based on Ring Segmentation

Ring segmentation [11] is introduced to process the second-level CS-LBP texture image. Pixels are assigned to rings based on their distance from the image center. Let R_i denote the pixel set of the i-th ring. Statistical features [12] (mean, variance, skewness, and kurtosis) are extracted to represent the characteristics of pixels within each ring, effectively converting high-dimensional features into low-dimensional representations while increasing robustness. Let n be the number of rings, r_k be the k-th ($k = 1, 2, ..., n$) ring radius, and r_n can be calculated using equation (3):

$$r_n = \left\lfloor \frac{M}{2} \right\rfloor$$

where $\lfloor \cdot \rfloor$ denotes floor operation. Other ring radii are calculated based on the inscribed circle area S_0 and mean area $\bar{S}$. The ring radius calculations are shown in equations (5) and (6):

$$S_0 = \pi r_1^2, \quad \bar{S} = \frac{\pi r_n^2}{n}$$

$$r_k = \sqrt{\frac{(k-1)\bar{S}}{\pi} + r_1^2}, \quad k = 2, 3, ..., n$$

The distance d from a pixel location to the image center is:

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

where (x_c, y_c) are the image center coordinates and (x, y) are pixel coordinates.

Pixel membership in each set is determined by equations (8) and (9):

$$R_1 = \{(x,y) | d(x,y) \leq r_1\}$$

$$R_k = \{(x,y) | r_{k-1} < d(x,y) \leq r_k\}, \quad k = 2, 3, ..., n$$

The mean, variance, skewness, and kurtosis of each ring are calculated to represent the corresponding ring features. n rings yield a $4 \times n$ dimensional feature matrix V. Matrix V is standardized row-wise. Let the mean of the i-th row of feature matrix V be the reference vector C_i. The distance from elements in feature matrix V to the reference vector is used as an invariant feature, as shown in equation (10):

$$D_{i,j} = |V_{i,j} - C_i|$$

where i and j represent row and column indices, respectively.

The resulting $4 \times n$ dimensional invariant feature matrix D is unfolded column-wise to obtain a $1 \times 4n$ dimensional feature vector D_1 as the texture feature. Similarly, the feature vector D_2 for the third-level CS-LBP texture image is obtained, resulting in the final feature vector $H_{CS} = [D_1, D_2]$.

1.6 Statistical Features Based on Bit Planes

Noise attacks significantly affect high-frequency information obtained from wavelet decomposition. However, after bit plane decomposition, noise information concentrates in low-order bit planes while high-order bit planes retain important information [13]. Based on this characteristic, bit plane theory is applied to the horizontal and vertical components of high-frequency information from the second-level wavelet decomposition. The statistical histograms of bit planes 5, 6, 7, and 8 are extracted and normalized to obtain an 8-dimensional feature vector H_W. The same process is applied to high-frequency information from the third-level wavelet decomposition to obtain another 8-dimensional feature vector H_{W1}. The final 16-dimensional feature vector $H_{WPM} = [H_W, H_{W1}]$ effectively improves robustness to noise while ensuring distinctiveness between different images.

1.7 Hash Generation

To demonstrate the effect of bit images, the low-frequency and high-frequency features are combined at a ratio of 1:4, as shown in equation (11), followed by rounding operation in equation (12):

$$H = 4 \times H_{CS} + H_{WPM}$$

$$H = \text{round}(H + 0.5) \times 10$$

From Sections 1.4 and 1.5, the hash length is $8n + 16$ integers. A random generator produces a pseudo-random number sequence K of length $8n + 16$ as the key. The hash sequence is scrambled using key K as shown in equation (13):

$$h(i) = H(K(i))$$

where i represents the i-th number in sequence K.

1.8 Image Copy Detection

For an image database to be detected and a query image, the corresponding hash sequence database and query image hash sequence are obtained through the hash function. The hash distance between the query image hash sequence and hash sequences in the database is calculated using Euclidean distance, as shown in equation (14):

$$D(h_1, h_2) = \sqrt{\sum_{i=1}^{N} (h_1(i) - h_2(i))^2}$$

When the Euclidean distance is less than a set threshold, the image is determined to be a copied image. The copy detection process is illustrated in [Figure 2: see original paper].

2 Experimental Results and Analysis

Experimental parameters are set as follows: normalized image size $M = 256$, 3×3 Gaussian low-pass filter with standard deviation 3, CS-LBP radius $R = 1$, number of pixel points $N = 8$, number of rings $n = 40$, resulting in a hash length of 336 integers.

2.1 Robustness Experiment

In the robustness experiment, five test images (Airplane, House, Lena, Baboon, and Peppers) were used as samples. Image processing parameters are shown in . Experimental results are presented in [Figure 3: see original paper].

The horizontal axis in [Figure 3: see original paper] represents conventional processing (from), and the vertical axis represents the hash distance between processed images and original images. The results show that conventional processing has minimal impact on the hash, except for rotation operations. For rotations greater than 1°, the hash distance increases rapidly. This is because the conversion of approximation images to CS-LBP texture images makes the algorithm less robust to rotations above 1°. However, [Figure 3: see original paper] demonstrates that the proposed method exhibits good robustness to conventional processing except for large-angle rotations.

2.2 Distinctiveness Experiment

To verify the distinctiveness of the proposed hash algorithm, 700 images were extracted from the University of Washington Ground Truth Database [14] and 300 images from the VOC2007 database [15] to construct a database of 1,000 different images. The distinctiveness was verified on the MATLAB platform. There are 499,500 different image pairs among the 1,000 images. By applying the processing in to the 1,000 images, 105,000 similar image pairs were generated. Distinctiveness experiment results are shown in [Figure 4: see original paper], where red (see electronic version) represents the hash distance distribution of different image pairs and blue represents that of similar image pairs. The figure shows a clear boundary between the two distributions, allowing an appropriate threshold to be selected for discrimination.

Error detection rate and collision rate formulas [16] are introduced as equations (15) and (16):

$$P_E = \frac{N_E}{N_S}, \quad P_C = \frac{N_C}{N_D}$$

where N_E is the number of similar image pairs detected as different, N_C is the number of different image pairs detected as similar, N_S and N_D are the total numbers of similar and different image pairs respectively, P_E is the error detection rate, and P_C is the collision rate.

The relationship between P_C and P_E with threshold is shown in . The results indicate that collision rate and error detection rate are mutually constrained, reflecting the trade-off between robustness and distinctiveness. In practice, an appropriate threshold can be selected based on requirements. According to , a threshold of 115 can be chosen, at which the algorithm achieves low collision and error detection rates.

2.3 Security Analysis

To verify algorithm security, a correct key was first used to extract the hash of the Lena image. Then, under otherwise identical conditions, 1,000 incorrect keys were used to obtain corresponding hashes. The distances between these incorrect-key hashes and the correct-key hash are shown in [Figure 5: see original paper].

All distances for incorrect keys exceed 150, significantly higher than the set threshold of 115. This demonstrates that correct hash sequences cannot be obtained without knowing the correct key, indicating good security of the proposed algorithm.

2.4 Parameter Discussion

This section discusses the impact of ring number and the ratio of H_{CS} to H_{WPM} on algorithm performance. Ring numbers of $n = 10, 20, 30, 40$ and ratios of 1:2, 1:4, 1:6, 1:8, 1:10 were tested using the 1,000-image database from Section 2.2 (499,500 different image pairs and 105,000 similar image pairs from processing in).

2.4.1 Impact of Ring Number With the H_{CS} to H_{WPM} ratio fixed at 1:4, ring numbers of 10, 20, 30, and 40 were tested, corresponding to hash lengths of 96, 176, 256, and 336 integers respectively. ROC curves [17] were used to compare performance. [Figure 6: see original paper] shows ROC curves for different ring numbers, where the horizontal axis represents false positive rate (FPR) reflecting distinctiveness, and the vertical axis represents true positive rate (TPR) reflecting robustness, calculated as:

$$FPR = \frac{N_{false}}{N_1}, \quad TPR = \frac{N_{true}}{N_2}$$

where N_{false} is the number of different image pairs judged as similar, N_{true} is the number of similar image pairs correctly identified, and N_1, N_2 are the total numbers of different and similar image pairs respectively.

For fair comparison, all experiments were conducted on MATLAB 2016a with an Intel(R) Core(TM) i7-4720HQ CPU @ 2.6 GHz and 8 GB RAM. [Figure 6: see original paper] shows that the algorithm achieves better classification performance with 40 rings, which effectively demonstrates the algorithm' s capabilities.

2.4.2 Impact of Different Ratios With ring number fixed at 40, ratios of 1:2, 1:4, 1:6, 1:8, and 1:10 were tested. Results are shown in [Figure 7: see original paper]. The 1:10 ratio yields the worst performance, while 1:4 and 1:6 are similar. The enlarged view of the upper-left corner shows that 1:4 is slightly closer to the upper-left corner than 1:6, so the ratio of 1:4 is selected.

2.5 Performance Comparison

To demonstrate classification performance and efficiency, the proposed algorithm was compared with hashing algorithms from references [4], [18], [19], and [20]. For fairness, the same 1,000-image database was used (499,500 different image pairs and 105,000 similar image pairs from processing in), with other parameters set to their literature defaults and respective distance measurement methods used for similarity analysis. ROC curves were employed as the analysis tool.

[Figure 8: see original paper] shows ROC curve comparisons. The proposed algorithm outperforms the comparison algorithms in classification performance. The average time per hash function execution was obtained by recording total time consumed during uniqueness experiments, as shown in . While the algorithm in [4] is faster, its classification performance is far lower than the proposed algorithm. The algorithm in [18] has similar runtime to ours, while algorithms in [19] and [20] have lower runtimes.

3 Copy Detection Application

To test copy detection performance, copied images of Figure 9: see original paper were generated using MATLAB, Photoshop, and PhotoMagic. Digital operations included brightness adjustment, contrast adjustment, gamma correction, JPEG compression, image scaling, salt-and-pepper noise, 3×3 Gaussian low-pass filtering, multiplicative noise, watermark embedding, mosaic, and text addition. Parameter settings were: brightness +20, contrast +20, gamma correction value 1.25, JPEG compression quality factor 60, scaling factor 0.75, salt-and-pepper noise level 0.006, Gaussian low-pass filter standard deviation 0.6, multiplicative noise level 0.02, watermark transparency 0.7, mosaic cell size 50, and text size 72. These operations generated 11 copied images, which were added to the 1,000-image database to create an 1,011-image database.

The original image was used as the query image. shows the 13 images with smallest distances to the query image. shows the time required to query copied images from the database. All 11 copied images have distances less than 100. With a threshold of 115, all copied images can be accurately identified. Querying takes 198.669 s when the hash sequence database is not built, but only 0.338 s when it is pre-built. The hash extraction time for the query image is higher than the average algorithm runtime in Section 2.5 because the query image dimensions (3136×4281×3) are much larger than those in the test database. These results demonstrate that the proposed algorithm achieves good performance in copy detection applications.

4 Conclusion

This paper constructs a hashing algorithm using statistical texture information combined with bit image statistics of high-frequency information. Statistical

features of texture information achieve robustness and distinctiveness, while bit image theory applied to high-frequency information improves noise resistance. Experiments show that the proposed algorithm has short hash length, fast computation speed, low collision rate, and low error detection rate. Compared with existing hashing algorithms, it achieves better classification performance and can accurately detect all copied images in copy detection applications, showing good application prospects. Future work will focus on improving robustness to large-angle rotations and further enhancing computational speed.

References

[1] Kim C. Content-based image copy detection [J]. Signal Processing: Image Communication, 2003, 18(3): 169-184.

[2] Ling H, Cheng H, Ma Q, et al. Efficient image copy detection using multiscale fingerprints [J]. IEEE Multimedia, 2012, 19(1): 60-69.

[3] Wu K, Wang X, Yao J. Copy image detection based on Shi-Tomasi corner regions [J]. Journal of China Jiliang University, 2014, 25(3): 263-267.

[4] Tang Z, Yang F, Huang L, et al. Robust image hashing with dominant DCT coefficients [J]. Optik-International Journal for Light and Electron Optics, 2014, 125(18): 5102-5107.

[5] Tang Z, Yang F, Huang Z, et al. Image hashing algorithm based on PCA feature distance [J]. Journal of Guangxi Normal University: Natural Science Edition, 2016, 34(4): 9-18, 7.

[6] Li X, Xia X. Robust image hashing based on content structure graph [J]. Journal of Applied Sciences, 2016, 34(6): 691-701.

[7] Srivastava M, Siddiqui J, Ali M A. Robust image hashing based on statistical features for copy detection [C]//Proc of IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering. 2017: 490-495.

[8] Wang W, Huang F, Li J, et al. Face description and recognition using LBP pyramid [J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(1): 94-100.

[9] Tong M, Wang C, Zhang Z. Image hashing algorithm in DWT domain [J]. Journal of Shandong Normal University: Natural Science Edition, 2016, 31(2): 64-67.

[10] Heikkilä M, Pietikäinen M, Schmid C. Description of interest regions with center-symmetric local binary patterns [J]. Lecture Notes in Computer Science, 2006, 42(3): 58-69.

[11] Tang Z, Zhang X, Zhang S. Robust perceptual image hashing based on ring partition and NMF [J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(3): 711-724.

[12] Tang Z, Zhang X, Li X, et al. Robust image hashing with ring partition and invariant vector distance [J]. IEEE Transactions on Information Forensics & Security, 2015, 11(1): 200-214.

[13] Feng S, Zhong S, Yao X, et al. Research on noise in multiple bit planes of grayscale images under common watermark attacks [J]. Computer Technology and Development, 2012(11): 250-252.

[14] University of Washington. Ground truth database [EB/OL]. [2017-10-07]. http://imagedatabase.cs.washington.edu/groundtruth/.

[15] Redmon J. Pascal VOC dataset mirror [EB/OL]. (2012) [2017-10-07]. https://pjreddie.com/projects/pascal-voc-dataset-mirror/.

[16] Zhao Y, Wang S, Yao H, et al. Robust image Hash combining conformal mapping and Zernike moments [J]. Journal of Applied Sciences, 2012, 30(1): 75-81.

[17] Fawcett T. An introduction to ROC analysis [J]. Pattern Recognition Letters, 2006, 27(8): 861-874.

[18] Tang Z, Yang F, Huang L, et al. DCT and DWT based image hashing for copy detection [J]. ICIC Express Letters, 2013, 7(11): 2961-2967.

[19] Monga V, Mihcak M K. Robust and secure image hashing via non-negative matrix factorizations [J]. IEEE Transactions on Information Forensics & Security, 2007, 2(3): 376-390.

[20] Ou Y, Rhee K H. A key-dependent secure image hashing scheme by using radon transform [C]//Proc of International Symposium on Intelligent Signal Processing and Communication Systems. [S.l.]: IEEE Xplore, 2009: 595-599.

Note: Figure translations are in progress. See original paper for figures.