

Image Copy Detection Based on Local Binary Pattern and SVM Classifier

Mayank Srivastava¹, Jamshed Siddiqui², Mohd. Athar Ali³

¹Department of Computer Engineering & Applications, GLA University, Mathura, U.P., India

²Department of Computer Science, Aligarh Muslim University, Aligarh, U.P., India

³Department of Applied Computing, University of Buckingham, Buckingham, UK

E-mails: mayank.srivastava@gla.ac.in,

jamshed_faiza@rediffmail.com

athar.ali@buckingham.ac.uk.

Abstract: Due to the availability of a large number of image editing software, it is very easy to find duplicate copies of original images. In such a situation, there is a need to develop a robust technique that can be used for the identification of duplicate copies apart from differentiating it from different images. In this paper, we have proposed an image hashing technique based on uniform Local Binary Pattern (LBP). Here, the input image is initially pre-processed before calculating the Local Binary Pattern (LBP) which is used for image identification. Experiments show that proposed hashing gives excellent performance against the Histogram equalization attack. The Receiver Operating Curve (ROC) indicates that the proposed hashing also performs better in terms of robustness and discrimination. Support Vector Machine (SVM) classifier shows that generated features can easily classify images into a set of similar and different images, and can classify new data with a high level of accuracy.

Keywords: Content-based copy Detection, Digital watermarking, Local Binary Pattern, Support Vector Machine, Image forensics, Image hashing.

1. Introduction

The complexity of unauthorized replication and counterfeit techniques for multimedia content has always been ahead of copy detection and image forensic techniques [1]. The advent of cheap and sophisticated software allows us to do the unauthorized duplication of multimedia content in a very easy to use manner [2]. Due to this increase in unauthorized duplication, protecting the copyright of an image along with its identification is gaining importance [3]. In such a scenario, image authentication techniques are vital which is part of image forensics used to prove that the given image is original and is not a modified copy [4]. Digital watermarking is one of the earlier image forensic techniques in which a unique signature is appended to the image for identification. Content-Based Copy Detection (CBCD) techniques are an alternative to watermarking in which multimedia content based features are used for identification [5]. Recently, a significant amount of work is done in an

extended version of CBCD techniques named Image hashing [6]. In Image hashing, the extracted unique feature is represented in compressed form for identification [7]. Practically any image hashing mechanism should fulfil the properties related to robustness and discrimination apart from fulfilling the properties related to specific applications.

2. Literature review

In the past, researchers have implemented many algorithms related to various dimensions of image hashing. Some of the notable algorithms are as follows: Karsh, Laskar and Aditi [8] proposed an image hashing based on DWT-SVD (Singular value decomposition) to generate an image hash used for identification. Tang et al. [9] proposed an image hashing technique based on dominant DCT coefficients, which perform well in classification. Chen and Shang-Lin Hsieh [10] proposed an image hashing algorithm based on SIFT features. Ouyang, Coatrieux and Shu [11] propose the use of Quaternion Discrete Fourier Transform (QDFT) and Log Polar Transform (LPT) for image hashing, which shows good sensitivity to non-authorized image content alterations.

Liu et al. [12] proposed hashing based on wave atom transform, which outperforms DCT and DWT based techniques. Tang et al. [13] proposed an image hashing approach based on the color vector angle and canny edge operator. Tang et al. [14] proposed block-based robust image hashing based on color-vector angles and DWT, which is robust to normal digital operations including rotation up to 5° . Karsh, Laskar and Richhariya [15] proposed image hashing based on ring based Projected Gradient Non-negative Matrix Factorization (PG-NMF) features and local features. The method is robust to content preserving operations and is capable of localizing the counterfeit area. Chen, Yu and Feng [16] proposed image hashing based on Tchebichef moments, which gives superior robustness and discrimination values. Tang et al. [19] proposed technique based on Quaternion SVD which generates stable and discriminative image features. Tang et al. [20] proposed a robust hashing method based on NMF, which exhibits low collision probability. Tang et al. [21] proposed an approach based on visual attention model and invariant moments, which provides good rotation robustness.

At any case, all above-mentioned hashing algorithms attempt to address a variety of issues related to image hashing. More focus is required towards developing high-performance algorithms that perform well for different performance parameters particularly considering *Histogram equalization* attack. In this work, we propose a Local Binary Pattern (LBP) based novel image hashing technique that is robust to the *Histogram equalization* attack. The rest of the paper is arranged as follows: Section 3 describes the proposed image hashing technique. Section 4 gives the experimental results and the paper is concluded in Section 5.

3. Proposed Image Hashing

In this section, the various components used in the proposed technique are explained in detail.

3.1. Local Binary Pattern (LBP)

LBP is one of the most popular used features used in the field of texture classification, face analysis, and copy-move forgery. LBP is a binary code for an image-pixel around the local neighborhood of that pixel. If the neighboring pixel has a higher intensity as compared to the center pixel, then it is assigned value one otherwise zero is assigned. In such a manner LBP code for each of the center pixels is produced by combining previously generated binary code. LBP can also be extended to calculate its value from the neighborhood pixels of a circle, which can be given as notation (P, R) where P is the number of pixels in the neighborhood of the radius R . If the coordinates of the center pixel are (a_c, b_c) , then the coordinates of its P neighbors (a_p, b_p) on the edge of the circle with radius R can be calculated as $a_p = a_c + R \cos(2\pi p/P)$ and $b_p = b_c + R \sin(2\pi p/P)$ [17].

If the intensity of the center pixel is i_c and the intensity values of its neighbors are i_p with $p = 0, \dots, P-1$, then the texture T in the neighborhood of pixel (a_c, b_c) can be given as $T = t(i_c, i_0, \dots, i_{P-1})$. Since $t(i_c)$ describes the total luminance of an image, therefore, T can be rewritten as $T \approx (i_0 - i_c, \dots, i_{P-1} - i_c)$. T can also be modified as $T \approx (s(i_0 - i_c), \dots, s(i_{P-1} - i_c))$ where s represent the sign of component and $s(x)=1$ for $x \geq 0$ and $s(x)=0$ for $x < 0$. To produce the LBP for pixel (a_c, b_c) a binomial weight 2^p is assigned to each sign $s(i_p - i_c)$. These binomial weights are summarized as given in, (1). The total number of patterns for a uniform binary pattern is $P(P-1)+2$. For every pixel of an image, LBP code is calculated and the histogram of these patterns forms the feature vector. These feature vectors are finally used to measure the similarity between images by calculating the hash distance between the histograms.

$$(1) \quad \text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p.$$

3.2. Linear separable classification based on Support Vector Machine (SVM)

A SVM is a classification method based on discrimination of given labeled training data formally defined by a separating hyperplane. In two-dimensional space, the hyperplane can be defined as a line that divides the space into two parts where each part is represented by a particular class. If we have M training points where each input x_i has N attributes, then each point can be classified into one of two classes $y_i = -1$ or $+1$.

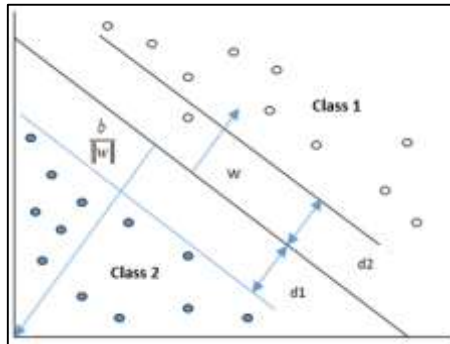


Fig. 1. SVM with two linearly separable classes

The training data can be written in the form of $\{x_i, y_i\}$ where $i = 1, \dots, M$, $y_i \in \{-1, +1\}$, $x \in R^N$ [18]. The hyperplane obtained after classification can be given as $w \cdot x + b = 0$, where w is normal to the hyperplane and $b/\|w\|$ is the perpendicular distance from the hyperplane to the origin. The main objective of SVM is to maximize the margin so that hyperplane can be far from support vectors.

3.3. Implementation approach

The algorithm for the proposed technique consists of various steps of pre-processing, transformation and hash generation, which is given in Fig. 2.

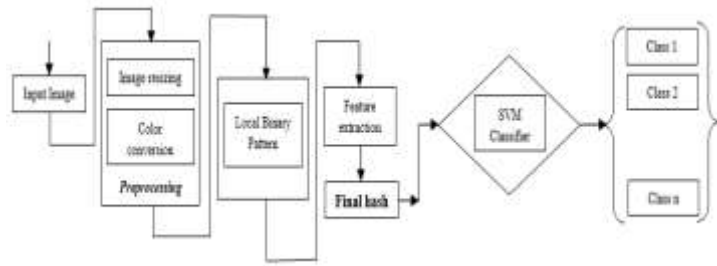


Fig. 2. Basic blockdiagram of the proposed hashing along with SVM classification

3.4. Image processing and Hash generation

The first step is normalization in which image resizing and color space conversion are performed. Image resizing is done to ensure that the input image is resized to a standard value of 512×512 . The gray space conversion of the resized image is done for hash generation. In the next step, uniform Local Binary Pattern (LBP) is generated for the processed image by dividing it into a cell size of 64×64 and considering the size of neighborhood pixels as 8. It is important here to clarify that we have considered a block size of 64×64 for generating a feature as it is found experimentally that using a smaller block size would lead to significant improvement in the result (Section 4.5).

3.5. Similarity metric

To measure the similarity between a pair of hashes, the Chi-square statistic is used. The hash distance between the two images hashes S and M can be given as follows:

$$(2) \quad \chi^2(S, M) = \sum_{j=1}^{64} \left(\frac{\sum_{i=1}^{P(P-1)+3} (S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}} \right)$$

The images being compared are considered identical if the result of hash distance is less than a predefined threshold, otherwise, they are considered as different images.

4. Experimental results

To demonstrate the efficiency of the proposed mechanism against different image-processing based attacks, a series of experiments have been conducted.

4.1. Robustness and Discrimination analysis

The proposed technique is applied to test images from the USC-SIPI image database [23]. The set of some of the standard images is used to create 67 modified copies by using a number of image processing operations which are used in most of the research papers of the area of image hashing [7, 9]. The duplicate copies were created by using MATLAB with the attack parameters shown in Table 1.

Table 1. Generation of duplicate copies of original images

| Attack | Details | Parameter variation | No of images |
|------------------------|--------------------|--|--------------|
| Brightness adjustment | Intensity values | 0.05, 0.10, 0.15, 0.20 | 4 |
| Contrast adjustment | Intensity values | 0.75, 0.80, 0.85, 0.90 | 4 |
| Cropping | X_{min}, Y_{min} | (2, 2), (4, 4), (6, 6), (9, 9), (11, 11) | 5 |
| Gamma correction | Gamma | 1.25, 1.5, 1.75, 2.0 | 4 |
| Gaussian Noise | Variance | 0.01, 0.02, 0.03, 0.04, 0.05 | 5 |
| Rotation | Angle | -1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1 | 8 |
| Salt & Pepper | Noise density | 0.01, 0.03, 0.05, 0.07, 0.09 | 5 |
| Speckle | Variance | 0.02, 0.04, 0.06, 0.08, 1.0 | 5 |
| Shift-H | Positions | 10, 20, 30, 40, 50 | 5 |
| Shift-V | Positions | 10, 20, 30, 40, 50 | 5 |
| Shift-HV | Positions | (10 10), (20 20), (30 30), (40 40), (50 50) | 5 |
| Histogram equalization | Gray levels | 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192 | 12 |
| Total | | | 67 |

Table 2. Maximum, minimum, mean and standard deviation of hash distances for different attacks

| Attack | Max | Min | Mean | Std Dev |
|------------------------|--------|-------|-------|---------|
| Brightness Adjustment | 66.83 | 12.98 | 35.46 | 14.73 |
| Contrast Adjustment | 102.39 | 12.33 | 29.43 | 18.82 |
| Cropping | 46.31 | 8.16 | 23.82 | 10.42 |
| Gamma Correction | 64.87 | 9.56 | 36.58 | 14.80 |
| Gaussian Noise | 64.22 | 9.26 | 34.31 | 15.24 |
| Rotation | 46.41 | 9.91 | 28.71 | 10.40 |
| Salt & Pepper | 77.07 | 6.58 | 30.25 | 20.14 |
| Speckle | 92.83 | 8.39 | 38.51 | 23.63 |
| Shift-H | 65.81 | 14.09 | 38.81 | 12.84 |
| Shift-V | 104.4 | 15.76 | 43.25 | 20.15 |
| Shift-HV | 78.72 | 12.69 | 42.08 | 16.60 |
| Histogram equalization | 26.24 | 3.69 | 13.16 | 5.66 |

After making the “duplicate copies”, image hashes are calculated from all the images and hash distance is computed between the original and its corresponding duplicate copies. Table 2 presents the max, min, mean and standard deviation of hash distances for different attacks. The mean value of *Histogram equalization* is 13.16, whereas the mean values of all other attacks are almost 3 times greater than the mean value of *Histogram equalization*. Also, the max value of *Histogram equalization* is 26.24, which is almost three times more than the max values of all compared attacks. By looking at the above values, we can conclude that the proposed method gives excellent values for *Histogram equalization* as compared to other mentioned attacks. In this experiment, we have used 11 original images and created 67 copies of each of the original images to produce 737 copies. It is justified to choose a hash distance of

86 as the threshold on which the proposed technique can correctly identify 98.91 % as copied ones from similar copies.

To demonstrate discriminability, 651 different images of sizes ranging from 225×225 to 2144×1424 are collected from the USC-SIPI database [23]. The hash distance is calculated between these images by taking one of the images as a base one. The max, min, mean and standard deviation calculated on the basis of 650 hash distances are 223.4, 74.12, 135.8 and 28.8 respectively. If the threshold is 86, then 0.77% of different images are falsely identified as similar images. Also, the mean value of discrimination is 135.8, which is almost three times greater than the highest mean of robustness. Keeping in view the values of robustness and discrimination, the proposed hashing exhibits promising results.

4.2. Distribution of Hash distance

The distribution of the hash distance for both similar and dissimilar images is given in Figs 3 and 4, respectively. It is evident from the figures that the hash distance between similar images is below the identified threshold of 86, with a few exceptions. Also, the number of outliers in both the categories of (similar and dissimilar) images conforms to the true positive and false negative analysis performed for evaluating robustness and discrimination.

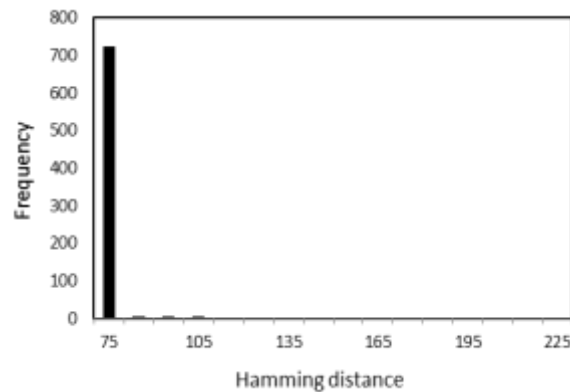


Fig. 3. Distribution of hash distance for similar images

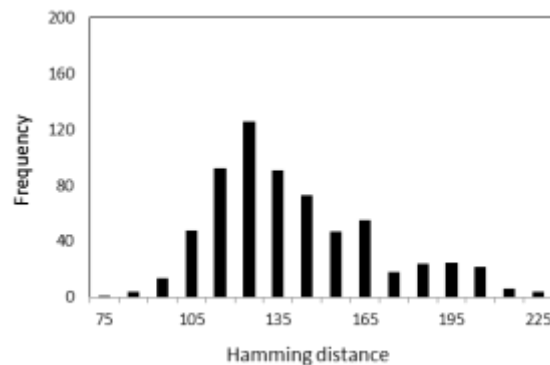


Fig. 4. Distribution of hash distance for different images

4.3. Performance comparison with state-of-the-art techniques

In this section, the ROC curve is plotted between True-Positive Rate (TPR) and False-Positive Rate (FPR) which is given in (3) where t_1 is the number of similar images correctly identified as copies and T_1 is the total number of similar images. Similarly, t_2 is the number of different images wrongly recognized as a copy and T_2 is the total number of different images,

$$(3) \quad \text{TPR} = \frac{t_1}{T_1}, \text{FPR} = \frac{t_2}{T_2}.$$

The proposed technique is compared by using the techniques given by [9, 20, 22]. To draw the ROC curve, it is important to calculate the above parameters for varying thresholds. The ROC curve for different algorithms including the proposed one is given in Fig. 5.

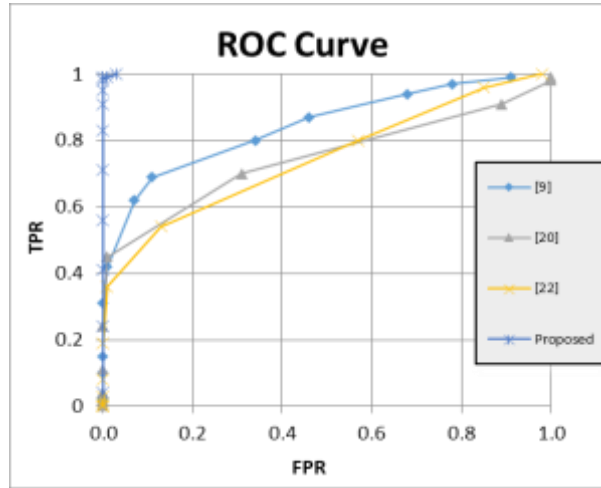


Fig. 5. ROC curve comparison between proposed and other hashing algorithms

From Fig. 5, it is evident that the ROC curve of the proposed technique is closer to zero as compared to the techniques reported in [9, 20, 22]. The values of TPR when FPR = 0 in case of [9, 20, 22] are 0.31, 0.19, & 0.24 respectively while for the proposed technique the value is 0.99. Similarly, the values of FPR when TPR = 1 in case of [9, 20, 22] are 0.91, 0.98 & 1.0 respectively while for the proposed technique the value is 0.03. Considering the TPR & FPR values obtained in this subsection, we can easily assert that the proposed technique performs well as compared to the notable hashing techniques.

4.4. Results of SVM classification

To perform classification, two image quality measures are used. The first measure is given in (2), which represents Chi-square statistic and the second measure is given in (4), which represents the Log-likelihood statistic. The values of both measures are calculated by using the dataset of similar and different images used in Section 4.1. SVM is trained by the values of these two measures along with their label to improve the classification results. Figs 6 and 7 represents the classification results based on

the SVM classifier by using linear and polynomial kernels respectively. One can conclude from the classification results that the proposed technique easily classifies between similar and different images.

$$(4) \quad \chi^2(S, M) = \sum_{j=1}^4 \left(- \sum_{i=1}^{P(P-1)+3} S_{i,j} \log M_{i,j} \right).$$

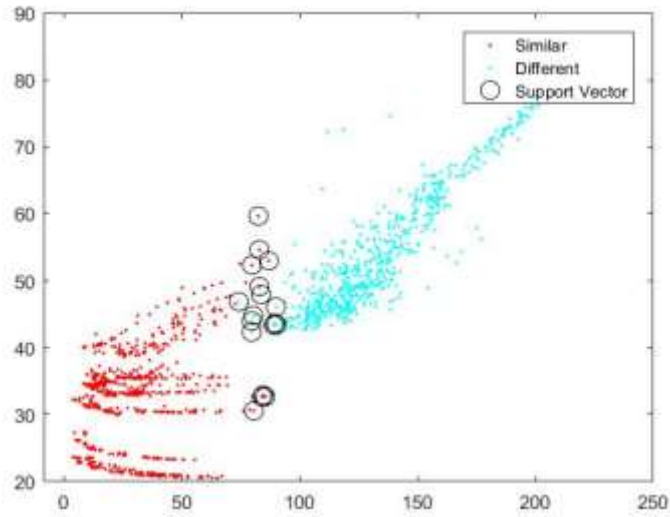


Fig. 6. Classification results based on SVM-Linear

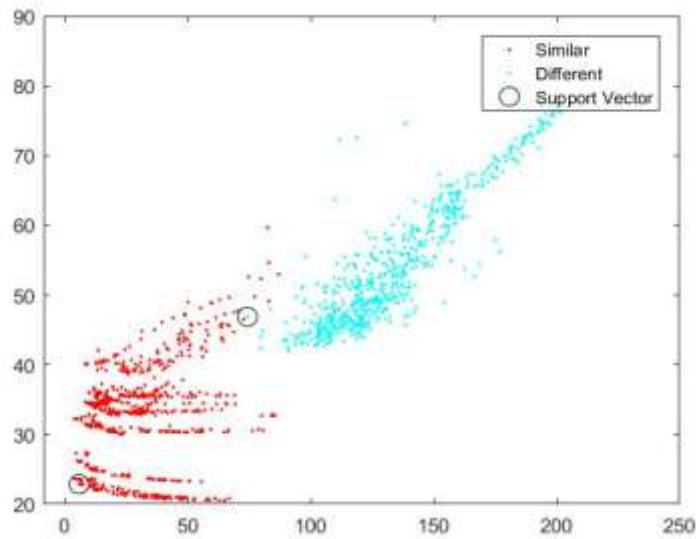


Fig. 7. Classification results based on SVM-Polynomial

Apart from the classification, the prediction of new data based on trained classifier is also done. The posterior probabilities for the predicted values are given in Table 3. It is clear from Table 3 that the proposed classifier gives excellent results pertaining to prediction values of similar and different images.

Table 3. Summary of prediction values using classified SVM

| Input data | True label | Predicted label | Linear Kernel | | Polynomial Kernel | |
|------------|------------|-----------------|--------------------------|---------------------------|--------------------------|---------------------------|
| | | | Posterior prob. similar | Posterior prob. different | Posterior prob. similar | Posterior prob. different |
| [17 40] | Similar | Similar | 1.0000 | 6.4584×10^{-10} | 0.9997 | 2.7814×10^{-4} |
| [16 35] | Similar | Similar | 1.0000 | 4.4423×10^{-10} | 0.9992 | 7.8631×10^{-4} |
| [38 39] | Similar | Similar | 1.0000 | 5.0058×10^{-7} | 0.9994 | 6.3703×10^{-4} |
| [15 69] | Similar | Similar | 1.0000 | 4.7611×10^{-10} | 1.0000 | 9.9327×10^{-10} |
| [51 45] | Similar | Similar | 1.0000 | 3.3174×10^{-5} | 0.9996 | 3.7636×10^{-4} |
| [66 20] | Similar | Similar | 0.9971 | 0.0029 | 0.7879 | 0.2121 |
| [135 49] | Different | Different | 7.6258×10^{-8} | 1.0000 | 1.6090×10^{-11} | 1.0000 |
| [171 80] | Different | Different | 5.8553×10^{-13} | 1.0000 | 0 | 1 |
| [204 79] | Different | Different | 0 | 1 | 0 | 1 |
| [214 86] | Different | Different | 6.2659×10^{-12} | 1.0000 | 0 | 1 |
| [180 40] | Different | Different | 5.3069×10^{-14} | 1.0000 | 0 | 1 |
| [110 30] | Different | Different | 2.6399×10^{-4} | 0.9997 | 3.6648×10^{-6} | 1.0000 |

We have also calculated the classification loss of SVM for different kernels, which can be defined as measuring the predictive inaccuracy of classification models. Table 4 below shows the loss value for every fold of the classification. It is clear from the given values that linear classification gives good results as compared to the polynomial kernel.

Table 4. Classification loss values

| Kernel /Fold | Loss values | | | | | | | | | |
|--------------|-------------|--------|--------|--------|--------|--------|--------|--------|-------|--------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Linear | 0.0072 | 0.0071 | 0.0071 | 0 | 0.0071 | 0.0071 | 0 | 0.0215 | 0 | 0.0144 |
| Polynomial | 0.0144 | 0.0071 | 0.0071 | 0.0071 | 0.0143 | 0.0071 | 0.0143 | 0 | 0.028 | 0 |

4.5. Comparing different variants of the proposed technique

This section verifies the effect of different cell sizes for LBP calculation by keeping the rest of the parameters constant. To perform the experiment, a set of 300 images have been taken from 17 Category Flower dataset [24] and further 67 “copies” were produced from a single “Airplane” image after applying various image processing operations (Table 1). Initially, the block size of 64×64 is used for LBP feature generation. In the subsequent steps, block size is increased to 128×128 , 256×256 and 512×512 respectively. To effectively represent the results, the ranking of the results based on the hash distance is done. The calculated rank for the first 62 copies of the original image at a block size of 64×64 , 128×128 , 256×256 and 512×512 are 62, 63, 65 and 75 respectively. One can easily conclude from the given values that lower block size generates a lower rank. Therefore, it gives the perfect reason to use the block size of 64×64 for our implementation. It is important here to mention that the result of 67 copies is not considered in this analysis as due to outliers we were getting

higher ranks for all the different block sizes. However, such result conforms to the results obtained for 62 copies.

5. Conclusion

In this paper, we have presented a robust image hashing technique that employs Local Binary Pattern (LBP) and Support Vector Machine (SVM) to differentiate duplicate copies of images from their original ones. The proposed technique shows remarkable robustness to different attacks apart from performing well specially against *Histogram equalization*. The proposed technique is compared against a number of state-of-the-art techniques by using the ROC curve, which shows that the proposed approach has better classification results. The proposed technique is also evaluated to know the effect of different block-size of LBP, which shows that small block sizes gives better results as compared to large block sizes. Finally, one can say that the proposed method can be used as a content-based image authentication in large-scale image databases.

References

1. Qureshi, M. A., M. Deriche. A Bibliography of Pixel-Based Blind Image Forgery Detection. – Signal Processing: Image Communication, Vol. **39**, 2015, pp. 46-74.
2. Liu, G., J. Wang, S. Lian, Z. Wang. A Passive Image Authentication Scheme for Detecting Region-Duplication Forgery with Rotation. – Journal of Network and Computer Applications, Vol. **34**, 2011, pp. 1557-1565.
3. Qazi, T., K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan, J. Kolodziej, H. Li, W. Lin, K. C. Yow, C. Z. Xu. Survey on Blind Image Forgery Detection. – IET Image Processing, Vol. **7**, 2013, pp. 660-670.
4. Battiato, S., G. M. Farinella, E. Messina, G. Puglisi. Robust Image Alignment for Tampering Detection. – IEEE Transactions on Information Technology Forensics and Security, Vol. **7**, 2012, pp. 1105-1117.
5. Khelifi, F. Perceptual Image Hashing Based on Virtual Watermark Detection. – IEEE Transactions on Image Processing, Vol. **19**, 2010, pp. 981-993.
6. Tang, Z., L. Chin, H. Yao, X. Zhang, C. Yu. Video Hashing with DCT and NMF. – The Computer Journal, 2019.
7. Tang, Z., X. Zhang, X. Dai, J. Yang, T. Wu. Robust Image Hash Function Using Local Color Features. – International Journal of Electronics and Communications (AEU), Vol. **67**, 2013, pp. 717-722.
8. Karsh, R. K., R. H., Laskar, and Aditi. Robust Image Hashing through DWT-SVD and Spectral Residual Method. – EURASIP Journal on Image and Video Processing, Vol. **31**, 2017, pp. 1-17.
9. Tang, Z., F. Yang, L. Huang, X. Zhang. Robust Image Hashing with Dominant DCT Coefficients. – Optik-International Journal for Light and Electron Optics, Vol. **125**, 2014, pp. 5102-5107.
10. Chen, C. C., Shang-Lin Hsieh. Using Binarization and Hashing for Efficient SIFT Matching. – Journal of Visual Communication and Image Representation, Vol. **30**, 2015, pp. 86-93.
11. Ouyang, J., G. Coatrieux, H. Shu. Robust Hashing for Image Authentication Using Quaternion Discrete Fourier Transform and Log-Polar Transform. – Digital Signal Processing, Vol. **41**, 2015, pp. 98-109.

12. Liu, F., L. Cheng, H. Leung, Q. Fu. Wave Atom Transform Generated Strong Image Hashing Scheme. – Opt. Communication, Vol. 7, 2012, pp. 5008-5018.
13. Tang, Z., L. Huang, X. Zhang, H. Liao. Robust Image Hashing Based on Color Vector Angles and Canny Operator. – International Journal of Electronics and Communications (AEU), Vol. 70, 2016, pp. 833-841.
14. Tang, Z., Y. Dai, X. Zhang, L. Huang, Fan Yang. Robust Image Hashing via Colour Vector Angles and Discrete Wavelet Transform. – IET Image Processing, Vol. 8, 2014, pp. 142-149.
15. Karsh, R. K., R. H., Laskar, B. B. Richhariya. Robust Image Hashing Using Ring Partition-PGNMF and Local Features. – SpringerPlus, Vol. 5, 2016, pp. 1-20.
16. Chen, Y., W. Yu, J. Feng. Robust Image Hashing Using Invariants of Tchebichef Moments. – Optik, Vol. 125, 2014, pp. 5582-5587.
17. Ojala, T., M. Pietikainen, D. Harwood. A Comparative Study of Texture Measures with Classification Based on Feature Distributions. – Pattern Recognition, Vol. 29, 1996.
18. Ryu, S-J, H-Y Lee, I-W Cho, H-K Lee. Document Forgery Detection with SVM Classifier and Image Quality Measures. – Advances in Multimedia Information Processing PCM 2008, Vol. 5353, 2008, pp. 486-495.
19. Tang, Z., M. Yu, H. Yao, H. Zhang, C. Yu, X. Zhang. Robust Image Hashing with Singular Values of Quaternion SVD. – The Computer Journal, 2019.
20. Tang, Z., S. Wang, X. Zhang, W. Wei, S. Su. Robust Image Hashing for Tamper Detection Using Non-Negative Matrix Factorization. – Journal of Ubiquitous Convergence and Technology, Vol. 2, 2008, pp. 18-26.
21. Tang, Z., H. Zhang, C. M. Pun, M. Yu, C. Yu, Z. Xianquan. Robust Image Hashing with Visual Attention Model and Invariant Moments. – Mathematical Biosciences and Engineering, Vol. 16, 2019, Issue 5, pp. 6103-6120.
22. Ou, Y., K. H. Rhee. A Key-Dependent Secure Image Hashing Scheme by Using Radon Transform. – In: Proc. of 2009 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS'09), 2009, pp. 595-598.
23. USC-SIPI Image Database (2007).
<http://sipi.usc.edu/database>.
24. 17 Category Flower Dataset.
<http://www.robots.ox.ac.uk/~vgg/data/flowers/17/>

Received: 12.12.2019; Second Version: 16.03.2020; Accepted: 03.04.2020