Copy–Move Image Forgery Detection Using Gray-Tones with Texture Description

Anuja Dixit and Soumen Bag

Abstract Copy–move forgery is a well-known image forgery technique. In this image manipulation method, a certain area of the image is replicated and affixed over the same image on different locations. Most of the times replicated segments suffer from multiple post-processing and geometrical attacks to hide sign of tampering. We have used block-based method for forgery detection. In block-based proficiencies, image is parted into partially overlapping blocks. Features are extracted corresponding to blocks. In the proposed scheme, we have computed Gray-Level Co-occurrence Matrix (GLCM) for blocks. Singular Value Decomposition (SVD) is applied over GLCM to find singular values. We have calculated Local Binary Pattern (LBP) for all blocks. The singular values and LBP features combinedly construct feature vector corresponding to blocks. These feature vectors are sorted lexicographically. Further, similar blocks discovered to identify replicated section of image. To ensure endurance of the proposed methods, Detection Accuracy (DA), False Positive Rate (FPR), and F-Measure are calculated and compared with existing methods. Experimental results establish the validity of proposed scheme for precise detection, even when meddled region of image sustain distortion due to brightness change, blurring, color reduction, and contrast adjustment.

Keywords Copy–move image forgery · Feature extraction · Image forensics · Local binary pattern · Singular value decomposition

A. Dixit (✉) · S. Bag
Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India
e-mail: anu2010cse1@gmail.com

S. Bag
e-mail: bagsoumen@gmail.com

B. B. Chaudhuri et al. (eds.), *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, Advances in Intelligent Systems and Computing 1024, https://doi.org/10.1007/978-981-32-9291-8_7

1 Introduction

Copy–move forged [1] image is obtained by imitating a part of image and gluing it at different locations upon the same image as shown in Fig. 1. The principal motive behind such kind of forgery technique is to replicate the objects of image or to conceal the selective information rendered through image. Copy–move forgery follows above perception, because of which forged portion has a constitutional resemblance to rest of the image. As forged section has implicit resemblance to rest portion of the image which made detection of such kind of forgery a complex process. Copy–move forged images can be fabricated using image processing software without leaving any trace of forgery. Tampered region of the image may go through several geometrical and post-processing attacks to make detection of forgery a complicated task. Detection of copy–move forged images is integrative part of image forensics [2]. Significant amount of research work is practiced in this field. Fridrich et al. [3] designed pioneer technique for sleuthing copy–move forgery. Their method extracts discrete cosine transform (DCT) features of overlapping blocks. The features are sorted and resemblance among feature vectors is computed to detect fiddled areas. Alkawaz et al. [4] suggested a colligated algorithm that also uses DCT coefficients as features for distinct blocks of varying sizes. Both of the techniques possess high computational complexity and inappropriate identification of altered areas when manipulated areas of image suffer from post-processing operations. Zandi et al. [5] proposed a method utilizing adjustive similarity threshold. Their method utilized standard deviation of the image blocks for computation of threshold proportions. In [6], Lee et al. suggested a strategy employing histogram of orientated gradients (HOG) to spot tampered areas. Their algorithm performed well when altered image suffer from small degree rotations, blurring, brightness adjustment, and color reduction. Silva et al. [7] applied point of interest as well as blocks of pixels for forged region detection. They utilized voting process in multiscale space.

From the literature survey, we observed that copy–move forgery detection results may sustain false matches (which are incorrectly detected as forged even if primitively they are not forged region of image). To increase the detection accuracy and

(a) Original Image (b) Forged Image

Fig. 1 An instance of region duplication image forgery (tampered region is outlined with red color rectangle)

reduce false matches while forged image enduring several post-processing attacks (blurring, contrast modification, luminance change, color diminution, etc.) is an open-research subject to cultivate. To minimize the presence of fictitious matches, we have used lexicographical sorting with Euclidean distance calculation accompanied with computation of shift vector. For precise localization of forged region, we have used hybridized feature extraction technique in which we have combined singular values (obtained from decomposition over GLCM) with LBP features. Proposed approach is invariant to post-processing attacks and obtained improved detection accuracy as well as reduced false matches than former techniques.

GLCM [8] is a texture feature extraction technique, which represents the frequency of occurrence of one gray tone, in a delimitated spatial linear relationship with other gray tones in image. SVD [9] possess algebraic and geometric invariant properties. For a given matrix, unique singular values [10] are obtained which is useful in firm representation for image blocks. Most important data of image block is possessed by largest singular values (LSV) while small singular values are sensitive towards noise. LSV possess effective stability even when an image endure minor distortions. LBP [11] operator facilitates integrated description for a texture patch with structural and statistical features. LBP is highly discriminative texture operator [12] which analyze numerous patterns corresponding to each pixel with its neighborhood. The signed difference between center and neighboring pixel value is invariant to variation in mean luminance as well as change in grayscale.

This paper is structured as follows. Section 2 illustrates the proposed algorithm. Section 3 shows the experimental outcome validating the legitimacy of the proposed approach by utilizing evaluation metrics. Finally, the proposed study is concluded in Sect. 4.

2 Proposed Methodology

Figure 2 presents the fundamental framework followed for copy–move forgery detection. The proposed technique utilize following steps:

2.1 Preprocessing

1. **Conversion of color channel**:
   Initially, if the input image is RGB then it is converted to grayscale image.
2. **Division of image in overlapping blocks**:
   After preprocessing step, image is parted into partially overlapping blocks. If the gray scale image is of dimension $M \times N$ and block size used for division of image is $B \times B$ then total number of blocks are $(M - B + 1) \times (N - B + 1)$.

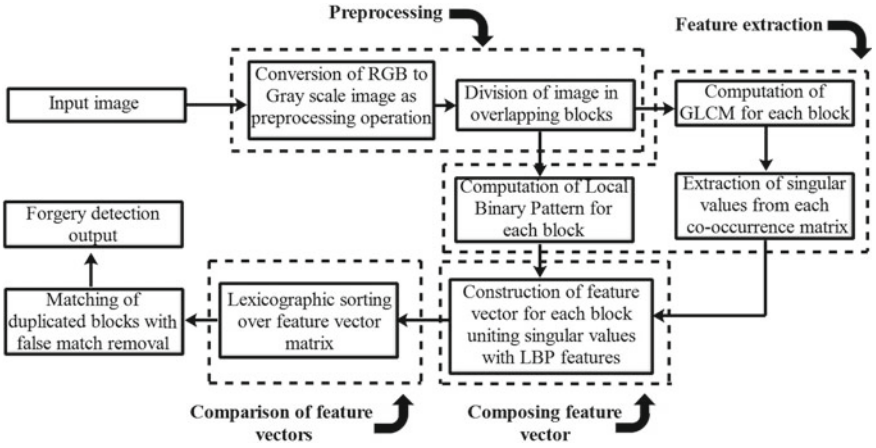

Fig. 2 Framework of the proposed methodology

2.2 Feature Extraction

1. **Computing gray-level co-occurrence matrix**:
   For each block, GLCM [8] is obtained to represent spatial relationship between the gray tones within each image block. The size of GLCM depends on maximum variation in gray level values of pixel within a block. For a block of dimension $B \times B$, co-occurrence matrix of size $B \times B$ is incurred.
2. **Calculation of singular values**:
   SVD [9] is applied over co-occurrence matrix. As result of decomposition, GLCM decomposed in three components: left unitary, right unitary, and diagonal matrix. Diagonal matrix contains singular values [10] on diagonal positions. Singular values represent feature vectors for all blocks of image. For a $B \times B$ dimensional matrix feature vector obtained is of length B.
3. **Extraction of LBP features and fusion with singular values**:
   Blocks are processed using LBP [11] as shown in Fig. 3. LBP features [12] are stored for each block. Singular values prevailed using SVD, and LBP features combinedly formulate feature vector to represent each block of image.

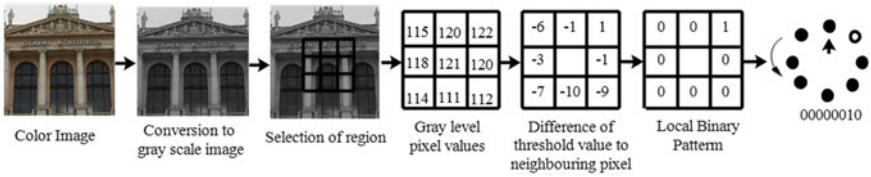

Fig. 3 Working rationale of local binary pattern operator

2.3 Comparison of Feature Vectors

Feature vectors for overlapping blocks are placed in matrix FM. The dimension of feature vector matrix FM is $(M - B + 1)(N - B + 1) \times len$, where '$len$' represents length of feature vector. For robust localization of similar feature vectors of duplicated blocks, Lexicographical sorting is applied over feature matrix. As a resultant, we achieve similar feature vector settled at neighboring locations.

$$FM = \begin{Bmatrix} FV_1 \\ FV_2 \\ FV_3 \\ \vdots \\ FV_{(M-B+1)(N-B+1)} \end{Bmatrix}$$

2.4 Similarity Detection and False Match Removal

1. **Duplicate region detection**:
   Euclidean distance is measured to find similarity between feature vectors as indicated in Eq. 2.

$$D(F, F') = (\sum_{i=1}^{n_f} (F_i - F_i')^2)^{\frac{1}{2}} \tag{2}$$

   where F and F′ shows feature vectors obtained from blocks and n_f represent the length of feature vector. To detect similar feature vectors from sorted feature matrix threshold value is used, such that $D(F, F') \leq T_{dis}$. To detect similar regions of image, shift vectors between similar blocks are measured. The top left corner location of a block is deliberated as it's emplacement within image. Let, (i_1, j_1) and (i_2, j_2) be the coordinates of blocks $\bar{FM}$. Shift vector between two co-ordinates can be obtained as in Eq. 3.

$$S = (s_1, s_2) = (i_1 - i_2, j_1 - j_2) \tag{3}$$

   Both Shift vectors $-S$ and S symbolize same order of magnitude for shifting. So, displacement vectors are normalized by multiplying with -1 to maintain $S \geq 0$. A counter value C is initialized with zero. For similar shift vector between neighboring blocks, counter value increased by 1 as in Eq. 4.

$$C(s_1, s_2) = C(s_1, s_2) + 1 \tag{4}$$

2. **Removal of false matches**:
   Counter value corresponding to different shift vector represents the frequency of similar shifting between block pairs. The proposed method detects all normalized shift vectors $S1, S2, S3, \ldots, Sr$. Shift vectors whose occurrence is higher than user-defined threshold T_{sh} shows forged blocks. $C(S_k) > T_{sh}$ for $k = 1, 2, \ldots, R$. High value of T_{sh} may leave altered regions undetected, while low value of T_{sh} can give rise to false matches. We employ a color map for localizing forged blocks of image.

3 Experimental Results and Discussion

3.1 Dataset

The source images for our experiments accumulated from CoMoFoD dataset [13]. Experiments are carried out on images with dimension 512×512. We selected 60 images suffering from color reduction, 60 images bearing contrast adjustment, 60 images with blurring attack, and 60 images with brightness change attack. In CoMoFoD dataset, color reduction images are divided into three categories with color channels contracted from 256 to 128, 64, and 32. Images with contrast adjustment also have three ranges [0.01, 0.95], [0.01, 0.9], and [0.01, 0.8]. Image with blurring attacks also have three categories based on average filter size as 3×3, 5×5, 7×7, and images with illumination change attack classified in three categories [0.01, 0.95], [0.01, 0.9], and [0.01, 0.8].

3.2 Experimental Setup

All experiments are executed over MATLAB 2016a, installed upon a platform equipped with 64-bit Windows (8 GB RAM) and Intel core i7 processor. We set every parameter as $B = 8$, $t_{dis} = 50$, and $t_{sh} = 50$. Various size of forged images can be used as input to proposed method, so values of M and N may vary. As here we consider all input images with size 512×512 so, $M = 512$ and $N = 512$. From experimental results, we found that when $B = 4$, too many false matches appear. when B increased to 16 then detection accuracy is compromised. T_{sh} is used for locating groups of blocks with similar shifting and meaningful forged region detection. t_{dis} is used for spotting similar feature vectors. Higher value of t_{dis} results in detection of different feature vectors as similar whereas smaller values of t_{dis} perform rigorous detection of similar feature vectors which may stipulate slightly dissident feature vectors as different. Here, length of extracted feature vector 'len' is 67.

3.3 Performance Evaluation

To demonstrate the outcomes of proposed technique, we have computed DA, FPR, and F-Measure by comparing the forgery detection outcomes with ground truth. DA, FPR, and F-Measure can be calculated as in Eqs. 5, 6, and 7 respectively.

$$DA = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

$$F\text{-}Measure = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

True Positive (TP) symbolizes the number of pixels correctly perceived as spoofed. False Positive (FP) represents the number of pixels which are falsely identified as faked but primitively they are not forged. False Negative (FN) shows the number of pixels which are manipulated but not detected. True Negative (TN) indicates number of pixels correctly detected as not forged. DA symbolizes the measure to which forgery detection algorithm can detect the altered region of image correctly, whereas FPR represents the percentage of pixels falsely detected as altered. F-Measure expresses the alliance of number of accurately detected forged pixels with respect to number of forged pixels perceived, and relevant forged region detection (i.e., proportion of correct detection of tampered pixels with respect to number of altered pixels acquainted by ground truth of forged image.).

Contrast Adjustment and Brightness Change Attacks

Table 1 illustrates average quantitative results obtained through the proposed method for forgery detection when images are enduring contrast adjustment and brightness change alterations. Qualitative outcomes incurred applying the proposed method are pictured in Fig. 4.

Table 1 Average DA and FPR results for contrast adjustment and brightness change attack

Contrast adjustment attack (Range)	DA (%)	FPR (%)	F-Measure (%)	Brightness change attack (Range)	DA(%)	FPR (%)	F-Measure (%)
[0.01, 0.95]	99.6647	1.0154	77.074	[0.01, 0.95]	97.8172	1.2816	68.269
[0.01, 0.90]	99.0486	2.1989	75.163	[0.01, 0.90]	95.5021	2.1989	67.725
[0.01, 0.80]	98.2311	2.6115	74.006	[0.01, 0.80]	94.9635	2.646	65.318

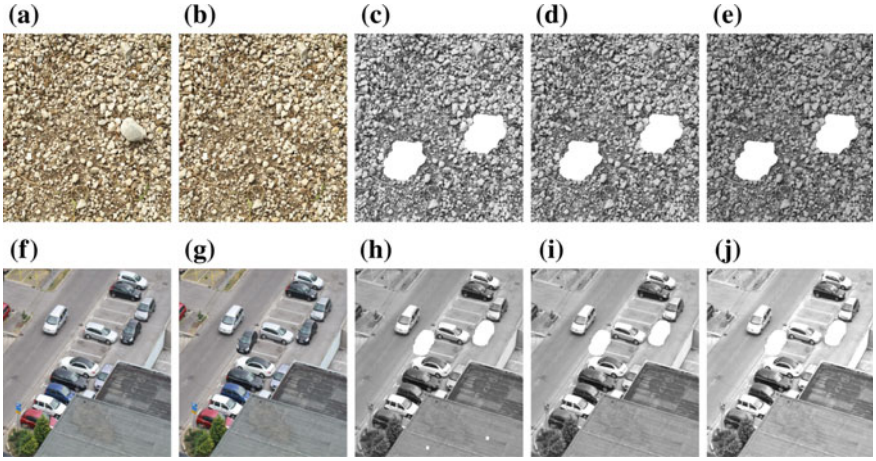

Fig. 4 **a**, **f** shows untampered image. **b**, **g** shows corresponding forged images. **c**, **d**, and **e** shows results for forgery detection in case of contrast modification category range as [0.01−0.8], [0.01−0.9], and [0.01−0.95] respectively. **h**, **i**, and **j** shows results for forgery detection in case of tampered images with luminance change range as [0.01−0.8], [0.01−0.9], and [0.01−0.95] respectively

Table 2 Average DA and FPR results for image blurring and color reduction attack

Image blurring attack (Filter size)	DA (%)	FPR (%)	F-Measure (%)	Color reduction attack (Level)	DA (%)	FPR (%)	F-Measure (%)
3 × 3	98.6736	0.9127	65.077	32	97.044	2.5684	64.810
5 × 5	97.0562	1.2478	64.008	64	97.8699	1.3162	67.295
7 × 7	96.7445	2.6115	61.016	128	98.8578	0.7159	71.627

Image Blurring and Color Reduction Attacks

Table 2 shows average quantitative results obtained using the proposed method for forgery detection when images are suffering from image blurring and color reduction attack. Qualitative results incurred using proposed method are shown in Fig. 5.

3.4 Comparative Analysis

Experimental outcomes establish the fact that our method is an efficacious technique for copy–move forgery detection. In comparison to other features based on DCT [3], SVD [11], HOG [6], and discrete wavelet transform (DWT) [14], our method

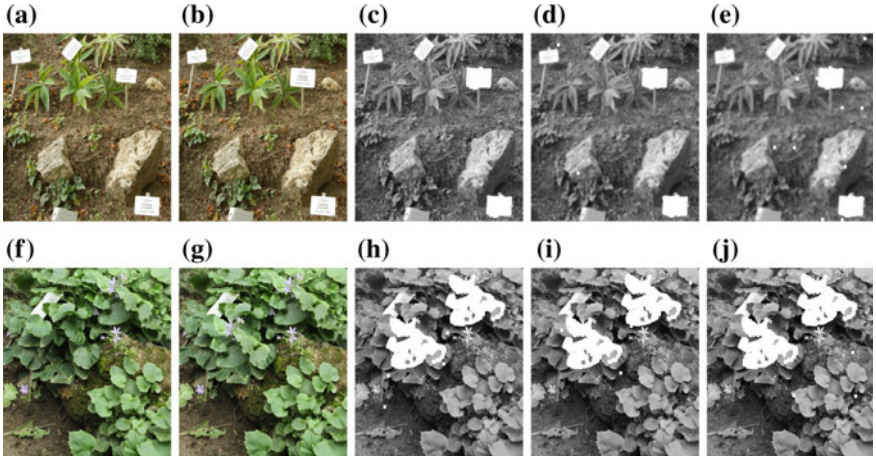

Fig. 5 **a**, **f** displays original image. **b**, **g** represents corresponding forged images. **c**, **d**, and **e** show results for forgery detection in case of image blurring with filter size 3×3, 5×5, and 7×7, respectively. **h**, **i**, and **j** shows results for forgery detection in case of color reduction with level 32, 64, and 128 respectively

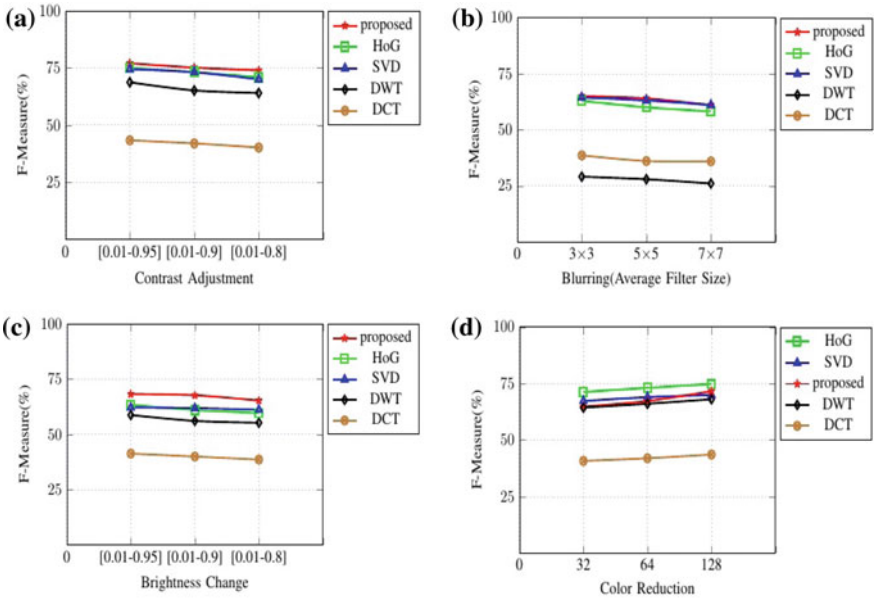

Fig. 6 **a**, **b**, **c**, and **d** portray comparative results for F-measure when image endures contrast adjustment, image blurring, brightness change, and color reduction attacks, respectively

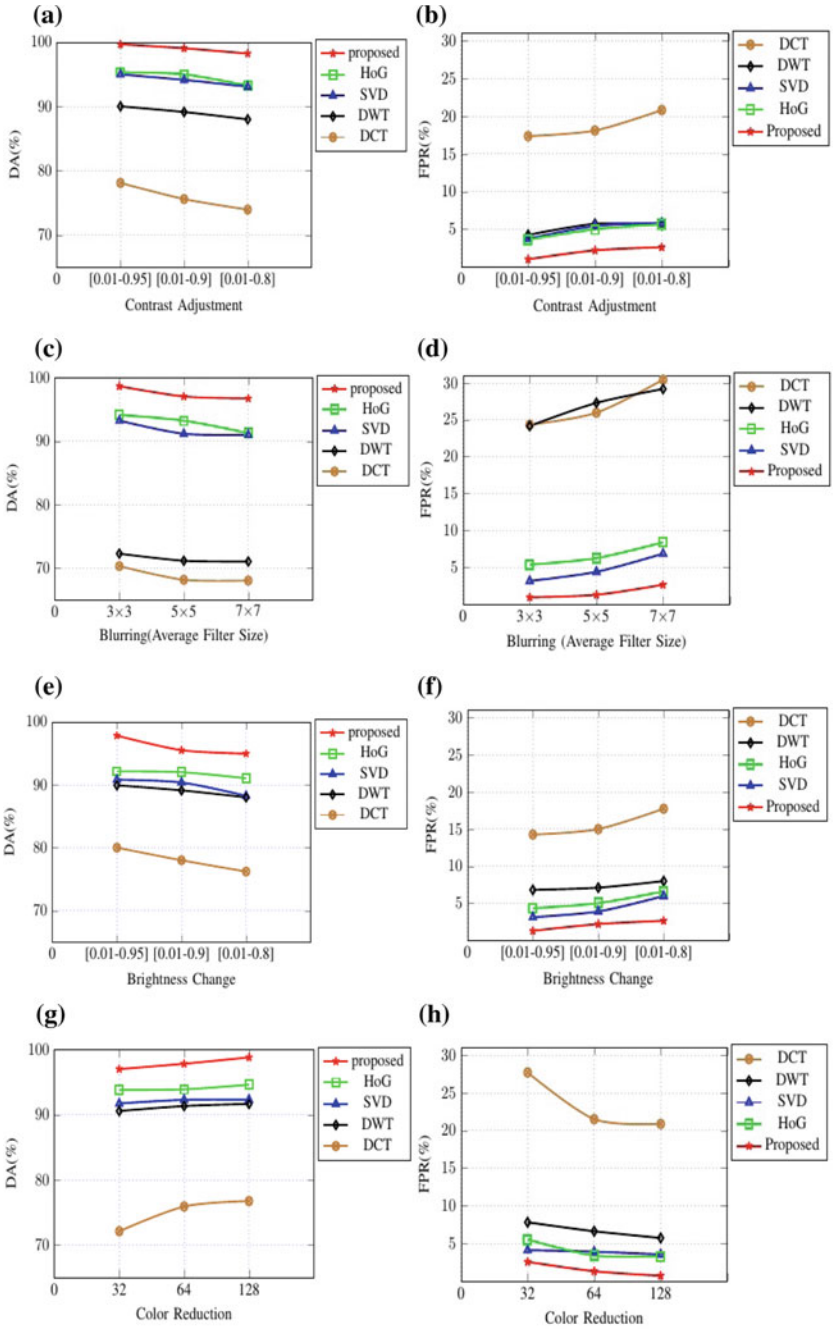

Fig. 7 **a**, **c**, **e**, and **g** show comparative results for DA, whereas **b**, **d**, **f**, and **h** represent comparative outcomes for FPR when forged image bears contrast adjustment, blurring, luminance change, and color diminution attacks, respectively

achieved highest DA and FPR rates for forgery detection, when altered image is suffering from contrast adjustment, image blurring, brightness change, and color reduction attacks. For color reduction attack, HoG-based features obtained highest F-measure for all color reduction levels. SVD features achieved better results for F-Measure than proposed method toward color reduction levels 32 and 64. Figure 6 displays comparative results obtained for F-measure when image endures post-processing attacks, whereas Fig. 7 shows comparative results for DA and FPR.

The complexity of copy–move forgery detection algorithms primarily depends on number of blocks accessed and length of feature vector generated, which are represented by n_b and n_f respectively. For input image size 512×512, the proposed algorithm obtains $n_f = 67$ and $n_b = 255{,}055$. DCT-based feature extraction method possess $n_f = 64$ and $n_b = 255{,}055$. For SVD-based method $n_f = 8$ and $n_b = 255{,}055$. Forgery detection scheme using HoG features obtains $n_f = 4$ and $n_b = 247{,}009$. DWT-based method acquires $n_b = 62{,}201$ and $n_f = 64$. For recording forgery detection time, we performed forgery detection operation over image size 256×256. The detection time for proposed algorithm is 98.65 s. For HoG, DWT, DCT, and SVD based methods detection times are 17.55 s, 1.69 s, 46.91 s, and 10.99 s respectively. Due to fusion of features obtained through singular value decomposition over GLCM and LBP, the proposed algorithm takes high computational time as compared to other cited methods, but facilitates high detection accuracy with low rate of occurrence of false matches.

4 Conclusion

Images have their applications in various fields like criminal investigation, courts of law, medical imaging, document analysis, etc. On account of speedy growth of technology, lot of potent computer applications are available which have made forging process easier. Forgery over images are practiced for denigration, blackmailing, harassment, political disputation, fun-making, etc. Digital images are not adequate in courts of law for witness, without forensic investigation over evidences. Such concerns related to multimedia security and forensic probes resulted in evolution of various advancements in image tampering detection techniques. As a little contribution in field of image forgery detection, our method obtained substantially high DA and low FPR when forged image suffer from blurring, color reduction, contrast adjustment, and brightness change attack. As future work, we will search for methods invariant to geometrical attacks with greater robustness than state-of-the-art methods of copy–move image forgery detection.

References

1. Zandi, M., Mahmoudi-Aznaveh, A., Talebpour, A.: Iterative copy-move forgery detection based on a new interest point detector. IEEE Trans. Inf. Forensics Secur. **11**(11), 2499–2512 (2016). https://doi.org/10.1109/TIFS.2016.2585118
2. Chen, C., Ni, J., Shen, Z., Shi, Y.Q.: Blind forensics of successive geometric transformations in digital images using spectral method: theory and applications. IEEE Trans. Image Process. **26**(6), 2811–2824 (2017). https://doi.org/10.1109/TIP.2017.2682963
3. Fridrich, J., Soukal, D., Lukas, J.: Detection of copy-move forgery in digital images. In: Digital Forensic Research Workshop, pp. 55–61. IEEE Computer Society (2003)
4. Alkawaz, M.H., Sulong, G., Saba, T., Rehman, A.: Detection of copy-move image forgery based on discrete cosine transform. Neural Comput. Appl. 1–10 (2016). https://doi.org/10.1007/s00521-016-2663-3
5. Zandi, M., Mahmoudi-Aznaveh, A., Mansouri, A.: Adaptive matching for copy-move forgery detection. IEEE International Workshop on Information Forensics and Security (WIFS), pp. 119–124 (2014). https://doi.org/10.13140/RG.2.1.2189.5200
6. Lee, J.C., Chang, C.P., Chen, W.K.: Detection of copy-move image forgery using histogram of orientated gradients. Inf. Sci. **321**(C), 250–262 (2015). https://doi.org/10.1016/j.ins.2015.03.009
7. Silva, E., Carvalho, T., Ferreira, A., Rocha, A.: Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. J. Vis. Commun. Image Represent. **29**, 16–32 (2015). https://doi.org/10.1016/j.jvcir.2015.01.016
8. Shen, X., Shi, Z., Chen, H.: Splicing image forgery detection using textural features based on grey level co-occurrence matrices. IET Image Process. **11**(1), 44–53 (2017). https://doi.org/10.1049/iet-ipr.2016.0238
9. Tai, Y., Yang, J., Luo, L., Zhang, F., Qian, J.: Learning discriminative singular value decomposition representation for face recognition. Pattern Recognit. **50**, 1–16 (2016). https://doi.org/10.1016/j.patcog.2015.08.010
10. Zhang, T., Wang, R.: Copy-move forgery detection based on SVD in digital images. In: International Congress on Image and Signal Processing, pp. 1–5 (2009). https://doi.org/10.1109/CISP.2009.5301325
11. Li, L., Li, S., Zhu, H., Chu, S.C., Roddick, J.F., Pan, J.S.: An efficient scheme for detecting copy-move forged images by local binary patterns. J. Inf. Hiding Multimed. Signal Process. **4**(1), 46–56 (2013)
12. Li, Z., Liu, G.Z., Yang, Y., You, Z.Y.: Scale and rotation-invariant local binary pattern using scale-adaptive texton subuniform-based circular shift and sub uniform-based circular shift. IEEE Trans. Image Process. **21**(4), 2130–2140 (2012). https://doi.org/10.1109/TIP.2011.2173697
13. Tralic, D., Zupancic, I., Grgic, S., Grgic, M.: CoMoFoD—new database for copy-move forgery detection. In: International Symposium Electronics in Marine, pp. 49–54 (2013)
14. Khan, S., Kulkarni, A.: An efficient method for detection of copy-move forgery using discrete wavelet transform. Int. J. Comput. Sci. Eng. **2**(5), 1801–1806 (2010)