Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8713202, 13 pages
http://dx.doi.org/10.1155/2016/8713202

Research Article

Copy-Move Forgery Detection Technique for Forensic Analysis in Digital Images

Toqeer Mahmood,[1] Tabassam Nawaz,[2] Aun Irtaza,[3] Rehan Ashraf,[1] Mohsin Shah,[4] and Muhammad Tariq Mahmood[5]

[1]*Department of Computer Engineering, University of Engineering and Technology, Taxila 47050, Pakistan*
[2]*Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan*
[3]*Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan*
[4]*Department of Information Technology, Hazara University, Mansehra 21140, Pakistan*
[5]*School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 330-708, Republic of Korea*

Correspondence should be addressed to Muhammad Tariq Mahmood; tariq@koreatech.ac.kr

Received 1 December 2015; Revised 16 April 2016; Accepted 9 May 2016

Academic Editor: Haipeng Peng

Due to the powerful image editing tools images are open to several manipulations; therefore, their authenticity is becoming questionable especially when images have influential power, for example, in a court of law, news reports, and insurance claims. Image forensic techniques determine the integrity of images by applying various high-tech mechanisms developed in the literature. In this paper, the images are analyzed for a particular type of forgery where a region of an image is copied and pasted onto the same image to create a duplication or to conceal some existing objects. To detect the copy-move forgery attack, images are first divided into overlapping square blocks and DCT components are adopted as the block representations. Due to the high dimensional nature of the feature space, Gaussian RBF kernel PCA is applied to achieve the reduced dimensional feature vector representation that also improved the efficiency during the feature matching. Extensive experiments are performed to evaluate the proposed method in comparison to state of the art. The experimental results reveal that the proposed technique precisely determines the copy-move forgery even when the images are contaminated with blurring, noise, and compression and can effectively detect multiple copy-move forgeries. Hence, the proposed technique provides a computationally efficient and reliable way of copy-move forgery detection that increases the credibility of images in evidence centered applications.

1. Introduction

With the advancements in imaging technologies, the digital images are becoming a concrete information source. Meanwhile, a large variety of image editing tools have placed the authenticity of images at risk. The ambition behind the image content forgery is to perform the manipulations in a way, making them hard to reveal through the naked eye, and use these creations for malicious purposes. For instance, in 2001, after the 9/11 incident, several videos of Osama bin Laden over the social media were found counterfeited through the forensic analysis [1]. In the same way, in 2007, an image of tiger in forest forced the people to believe in the existence of tigers in the Shanxi province of China. The forensic analysis, however, proved the tiger to be a "paper tiger" [2]. Similarly, in 2008, an official image of four Iranian ballistic missiles was found to be doctored, as one missile was revealed to be duplicated [3]. Hence, the famous saying "seeing is believing" [4, 5] is no longer effective. Therefore, ways that can ensure the integrity of the images especially in the evidence centered applications are required.

In recent years, an exciting field, digital image forensics, has emerged which finds the evidence of forgeries in digital images [6]. The primary focus of the digital image forensics is to investigate the images for the presence of forgery by applying either the active or the passive (blind) techniques [2]. The active techniques such as watermarking [7] and digital signatures [6] depend on the information embedded

(a) The original images (b) The copy-move forged images

Figure 1: An example of copy-move forgery.

a priori in the images. However, the unavailability of the information may limit the application of active techniques in practice [8]. Thus, passive techniques are used to authenticate the images that do not require any prior information about them [8–10].

Images are usually manipulated in two ways such as image splicing and region duplication through copy-move forgery. In image splicing, regions from multiple images are used to create a forged image. However, in copy-move forgery, image regions are copied and pasted onto the same image to conceal or increase some important content in the pictured image. As copied regions are apparently identical with compatible components (i.e., color and noise), it becomes a challenging task to differentiate the tempered regions from authentic regions. Furthermore, a counterfeiter applies various postprocessing operations such as blurring, edge smoothing, and noise to remove the visual traces of image forgeries. An example of copy-move forgery is shown in Figure 1.

In the present work copy-move forgery detection is addressed through the discrete cosine transform (DCT) and Gaussian RBF kernel PCA that are used to investigate the similarity between duplicated regions. The benefits of our algorithm compared against several existing CMFD methods are

(i) utilization of the lower length of feature vectors;

(ii) lower computational cost;

(iii) robustness against various postprocessing operations over the forged regions;

(iv) ability to detect multiple copy-move forgeries.

The rest of the paper is organized as follows: Section 2 presents the related work regarding copy-move forgery detection (CMFD). Section 3 presents the details of proposed method. Experimental results are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2. Related Work

Various CMFD techniques have been proposed so far to effectively address the region duplication problem. In this regard, the research is intended towards the representation of image regions in a more powerful way to accurately detect the duplicated regions. In [11], Fridrich et al. for the first time presented the copy-move forgery detection technique using DCT on small overlapping blocks. The feature vectors are formed using DCT coefficients. The similarity between blocks is analyzed after sorting the feature vectors lexicographically. In [13], image blocks are represented through principal component analysis (PCA). Exploiting one of the features of PCA, the authors used about half of the number of features utilized by [11]. It makes this technique effective but failed to detect copy-move forgery with rotation. In [15], a sorted neighborhood technique based on Discreet Wavelet Transform (DWT) is proposed. The image is decomposed into four subbands and applied the Singular Value Decomposition (SVD) on low frequency components for getting the feature vector. The technique is robust to JPEG compression up to the quality level 70 only. In [16], a technique based on blur moment invariants up to seventh order for extracting the block features and kd-tree matching is introduced. In [12], the application of scaling and rotation invariant Fourier-Mellin Transform (FMT) is suggested in combination with bloom filters on the image blocks for detecting the image forgery. In [14], an improved DCT-based technique is proposed by introducing a truncating process to reduce the dimension of feature vector for forgery detection. In [17], a solution through DCT and SVD is proposed for detecting image forgeries. The algorithm is shown to be robust against

compression, noise, and blurring but fails when images are even slightly rotated. In [18], an efficient expanding block technique based on direct block comparison is proposed. In [19], circle block extraction is performed and the features are obtained through rotation invariant uniform local binary patterns (LBP). The technique is robust to blurring, additive noise, compression, flipping, and rotation. However, this technique failed to detect forged regions rotated with arbitrary angles. In [20], the authors employed a new powerful set of keypoint-based features called MIFT for finding similar regions in the images. In [21], the authors extracted feature vectors from circular blocks using polar harmonic transform (PHT) for detecting image forgeries. In [22], an adaptive similarity threshold based scheme is presented in the block matching stage. The detection of forged regions is determined using thresholds proportional to blocks standard deviations. In [23], a method using the Histogram of Oriented Gradients (HOG) is suggested to detect the copy-move forged regions. In [24], the multiscale Weber's law descriptor (multi-WLD) and multiscale LBP features are extracted for image splicing and copy-move forgery detection from chrominance components. The authors employed SVM for classifying an image as authentic or forged.

3. Proposed Methodology

In this paper, copy-move forgery detection is performed through the DCT and Gaussian RBF kernel PCA using the squared blocks. The reason to use the DCT for block representation is the robustness against several postprocessing operations, for example, compression, blurring, scaling, and noise [25], as it is a common practice in image forgery that the counterfeited images always undergo various postprocessing operations. Hence, it makes the forgery detection very difficult. Although the DCT is effective against mentioned transformations, still there are situations where the block representations through DCT will be nominal; for example, if rotation operation is applied over the forged regions, the DCT representations results are affected as well. To overcome this limitation we apply Gaussian RBF kernel PCA over the DCT frequency coefficients due to their rotation invariant nature compared against PCA [25]. Another motivation to use kernel PCA with DCT is the nonlinear nature of RBF kernel PCA and linear nature of DCT. Hence, it makes the feature representation more diverse and also appears as a better choice compared to PCA that is also linear in nature like DCT. Gaussian RBF kernels have some other advantages such as having fewer hyperparameters; hence, they are numerically less difficult as kernel values are bounded between 0 and 1.

3.1. Framework of the Proposed Algorithm. The discussion above draws forth the framework of CMFD that is described in Figure 2. The steps of the proposed CMFD technique are given as follows:

(1) Dividing the grayscale image into fixed sized overlapping blocks.

(2) Applying DCT to each extracted block.

(3) Extracting Gaussian RBF kernel PCA-based features from each DCT square block.

(4) Matching similar block pairs.

(5) Removing the isolated block and output the duplicated regions.

3.2. Preprocessing and Blocks Extraction. For the implementation of proposed method, the algorithm is applied over the grayscale images. Thus, as a first step, a color input image I of size $H \times W$ is converted to a grayscale image using

$$I = 0.229R + 0.587G + 0.114B, \tag{1}$$

where R, G, and B are the red, green, and blue components of image I, respectively.

Once image I is converted into a grayscale image, a window of size $h \times w$ is slided one pixel along from the top left corner to the bottom lower right corner for obtaining the overlapping blocks. Each block is represented as B_{rc}, where r and c are the starting points of the block's row and column, respectively, as shown in

$$B_{rc}(x, y) = f(x + c, y + r), \tag{2}$$

where $x, y \in \{0, \ldots, B_{rc} - 1\}$, $r \in \{1, \ldots, H - h + 1\}$, and $c \in \{1, \ldots, W - w + 1\}$.

Thus, I can be divided into $\mathcal{N}$ overlapping blocks as shown in

$$\mathcal{N} = (H - h + 1) \times (W - w + 1). \tag{3}$$

3.3. Feature Extraction. For an image block $B_{rc}(x, y)$ of size $h \times w$, where x, y are $0, 1, 2, \ldots, N - 1$, we decompose the block $B_{rc}(x, y)$ in terms of 2D DCT basis function. The result occurs in the form of a coefficients matrix $\overline{C}(p, q)$ of size $h \times w$ that contains the DCT coefficients:

$$\overline{C}(p, q) = \alpha_p \alpha_q \sum_{x=0}^{h-1} \sum_{h=0}^{w-1} A_{hw} \cos \frac{\pi (2x + 1) p}{2h} \cos \frac{\pi (2y + 1) q}{2w}, \tag{4}$$

$$0 \le p \le h - 1, \ 0 \le q \le w - 1,$$

where

$$\alpha_p = \begin{cases} \dfrac{1}{\sqrt{h}}, & p = 0 \\ \dfrac{\sqrt{2}}{h}, & 1 \le p \le h - 1, \end{cases}$$
$$\alpha_q = \begin{cases} \dfrac{1}{\sqrt{w}}, & q = 0 \\ \dfrac{\sqrt{2}}{w}, & 1 \le q \le w - 1. \end{cases} \tag{5}$$

The coefficients matrix $\overline{C}(p, q)$ can be ordered to a zig-zag pattern to reflect the amount of information stored for block

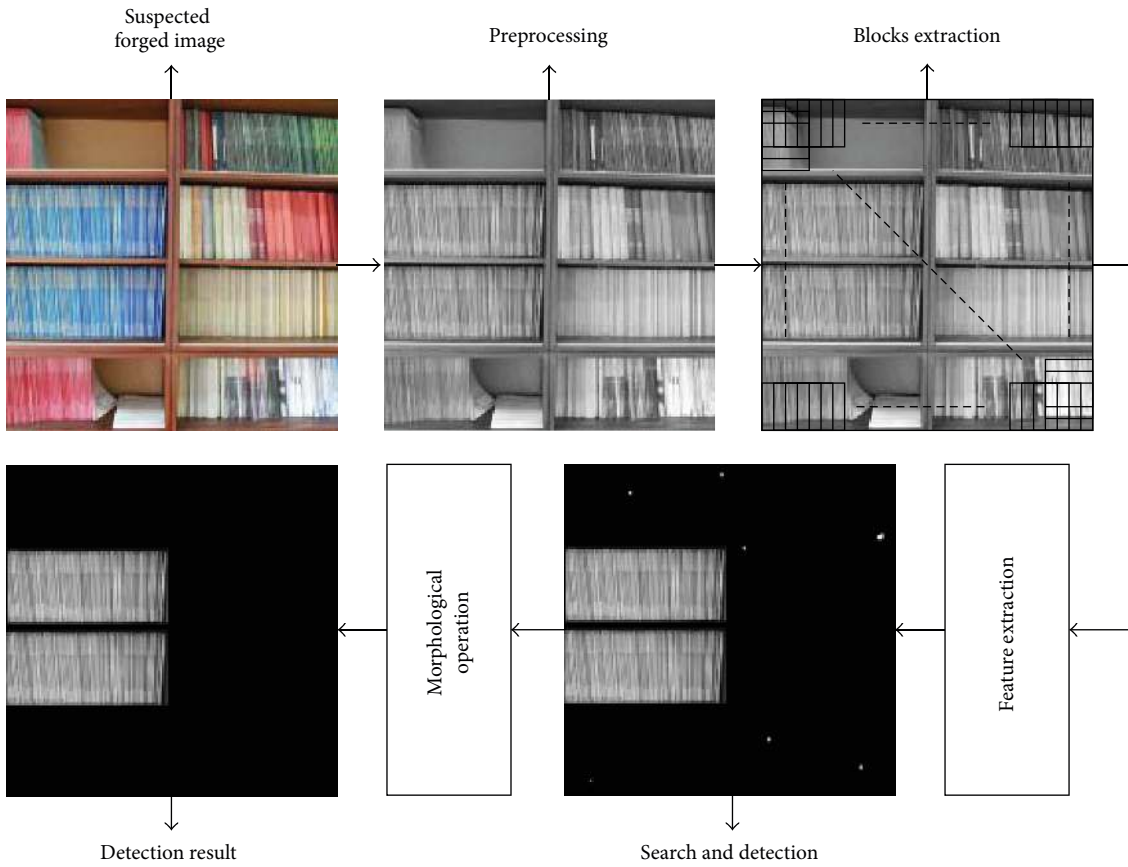

Figure 2: The proposed framework of the CMFD method.

representation. For a tempered block B_l that is a duplication of block B_k and does not undergo a postprocessing operation, the zig-zag coefficients are similar, and duplicated regions perfectly match but if geometric transformation such as rotation is applied on duplicated regions as

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix},$$
$$p' = p\cos\theta + q\sin\theta, \tag{6}$$
$$q' = -p\sin\theta + q\cos\theta,$$

where p' and q' represent the block coordinates after rotation with a rotation angle θ, the DCT coefficients may not match; that is,

$$\|d\|_2 = \sum_{i=1}^{n} \left| D_{l_i} - D_{k_i} \right| > \tau, \tag{7}$$

where $\|d\|_2$ is the distance between regions in metric norm. For region matching the value of $\|d\|_2$ must be less than a threshold τ. Therefore, to overcome this limitation of DCT coefficients, we compute the eigenvalues $\lambda \geq 0$ and eigenvectors $V \in F$ satisfying $\lambda V = CV$ with $C = \langle \phi(x_k)\phi(x_k)^T \rangle$, where ϕ is a nonlinear function. Thus, we can substitute C in the eigenvector equation. Therefore, all solutions V must lie in the span of ϕ. Equivalently,

$$\lambda\left(\phi\left(x_k\right) \cdot V\right) = \left(\phi\left(x_k\right) \cdot CV\right) \quad \forall k = 1, \ldots, l. \tag{8}$$

So there exist coefficients α_i such that

$$V = \sum_{i=1}^{M} \alpha_i \phi\left(x_i\right). \tag{9}$$

Substituting C and (9) in (8) and defining $M \times M$ matrix K by $K_{ij} \coloneqq (\phi(x_i) \cdot \phi(x_j)) = k(x_i, x_j)$ give

$$M\lambda\alpha = K\alpha, \tag{10}$$

where α represents the column vector and K is the symmetric Gram matrix [26] as given by

$$K\left(x_i x_j\right) = \phi\left(x_i\right)^T \phi\left(x_j\right). \tag{11}$$

If $\lambda_i \geq \lambda_{i+1}$ represents the eigenvalues of K and $i = 1, 2, \ldots, M$ and α^i denotes the eigenvectors, whereas V^i denotes the normalized eigenvectors provided $\lambda_k(\alpha^k \cdot \alpha^k) = 1$ for $\lambda_k \geq 0$, with an assumption that only the first p eigenvalues $\lambda' s$ are nonzero and positive. Therefore, for any

test point $\phi(x_t)$, the kth nonlinear principal components are given by [26]

$$\left\langle V^k, \phi(x_t)\right\rangle = \sum_{i=1}^{M} \alpha_i^k \left(\phi^T(x_i)\,\phi(x_t)\right) = \sum_{i=1}^{M} \alpha_i^k K(x_i, x_t), \quad \text{for } k = 1, 2, \ldots, p. \tag{12}$$

In our implementation, a Gaussian RBF kernel function is chosen, which is defined by the mapping function $\phi : [0, \infty) \to \Re$ such that

$$K\left(x, x'\right) = \phi\left(\left\|x - x'\right\|\right), \tag{13}$$

where $x, x' \in \Re^N$ and $\|\cdot\|$ represents the Euclidean distance. For any input space $\Re^N$, a kernel is a positive definite function for any integer n, satisfying $\sum_{i,j=1}^{M} \alpha_i \alpha_j K(x_i, x_j) \geq 0$, for any $\alpha_1, \ldots, \alpha_n \in \Re$ [26] and any $x_1, \ldots, x_n \in \Re^N$. Hence, the Gaussian RBF kernel is given as

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right), \tag{14}$$

where σ is the Gaussian kernel parameter. By applying (14) on (4), we get a transformed representation as

$$M_{\text{KPCA}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \cdots & a_{2L} \\ \vdots & \vdots & \cdots & \vdots \\ a_{Nc1} & a_{Nc2} & \cdots & a_{NcL} \end{bmatrix}. \tag{15}$$

The dimensionality of matrix M_{KPCA} can be reduced to N_C by using

$$1 - \varepsilon = \frac{\sum_{z=1}^{N_t} \lambda_z}{\sum_{z=1}^{b} \lambda_z}. \tag{16}$$

3.4. Block Representation Details. In our implementation, the size of each block B_i is $h \times w$, where the value of h and w is 16. When the DCT is applied to block B_i, we get a coefficients matrix $\overline{C}$ of the same size that is further transformed through the Gaussian RBF kernel PCA. By applying (16), we obtain first 10 most informative principal components that serve as the feature vector for the block representation. The reason to select these 10 principal components [27] is that we want to reduce the feature vector size due to large number of blocks and the curse of dimensionality due to which the time requirements for computation increase. Another reason is that the principal component analysis is orthogonal linear transformation such that the greatest variance by some projection appears as the first principal component, the second greatest variance appears as second principal component, and so on. In our implementation, the 10 most informative principal components are selected as the elements of feature vector for a block. Hence, a matrix M_{kpc} having the feature vectors of all the blocks is produced as given in

$$M_{\text{kpc}} = \begin{bmatrix} fv_1 \\ fv_2 \\ \vdots \\ fv_{(H-h+1)\times(W-w+1)} \end{bmatrix}. \tag{17}$$

The matrix M_{kpc} is sorted lexicographically which makes identical features closer to each other. In the meantime record the left corner's coordinate of each block that is represented by a square block. Lexicographical sorting before the feature vector matching procedure helps in decreasing the computational cost because a vector fv_i is compared against neighboring feature vectors N_n to judge the similarity. In our implementation, N_n window size is 20 to effectively handle various postprocessing operations such as blurring, noise, and compression.

Table 1 gives the comparison between the proposed method and other methods in terms of feature vector dimensions. In comparison with other methods our technique uses the lower dimension of feature vector and hence is computationally more efficient.

Table 1: Comparison of computational complexity.

Methods	Feature	Feature length
Fridrich et al. [11]	DCT	64
Bayram et al. [12]	FMT	45
Popescu and Farid [13]	PCA	32
Huang et al. [14]	Improved DCT	16
Proposed technique	DCT and KPCA	10

3.5. Forgery Detection. The target of the CMFD is to find duplicated regions where the similarity index between regions (feature vectors) is less than a certain threshold and the duplicated regions are nonoverlapping. The reason for the threshold based region matching is due to the nature of counterfeited images that undergo postprocessing operations; and the probability of being similar in terms of features is almost negligible. Therefore, for CMFD, two conditions are imposed over the duplicated block detection procedure: (1) the blocks are nonintersecting and nonoverlapping, and (2) the similarity index does not exceed a threshold.

To meet the first requirement of the CMFD, that is, the matching between nonoverlapping blocks, the shift distance criterion is used. For this, let us consider that (x_i, y_i) and (x_j, y_j) are the top left corner coordinates of the two blocks that are represented by the features vectors fv_i and fv_j, and then

$$\forall \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \geq N_t. \tag{18}$$

If the two feature vectors satisfy (18) then we consider these feature vectors for similarity index calculation to meet

the second requirement of the CMFD. For this Euclidean distance is adopted as

$$d\left(fv_i, fv_j\right) = \sqrt{\sum_{k=1}^{10}\left(fv_{ik} - fv_{jk}\right)^2} < d_t. \tag{19}$$

To show the result of forgery detection, the algorithm produces a black map image and the regions that are considered to be duplicated are highlighted as the desired output I_O.

3.6. Morphological Operations. To show the final detection result of the algorithm, morphological opening and closing operations at a scale defined by the size of the structural element are applied to I_O without losing any details of interest. Thus, the opening operation with a structural element of size 3×3 is used to remove the small and unwanted blocks of I_O, while closing operation with a structural element of size 8×8 is used to fill the holes in the highlighted regions of I_O.

3.7. Computational Analysis. For the clear description of process, we are adopting some standard notations as found in the computation text [28]. Suppose that the size of each block is $n \times n$ which was originally $h \times w$, where $h = w = 16$, and there are $\mathcal{N}$ total blocks. As a first step, we compute the 2D discrete cosine transform for a single block that takes $O(n \log n)$ time. Once the DCT is computed for a single block, we apply Gaussian RBF kernel PCA that takes $O(n^2)$ time [27]. Therefore, the machine instructions required for a single block is $O(n^2 + n \log n)$ and as in time complexity we are interested only in the largest factor; therefore, the time required to compute features for a single block is $O(n^2)$. As the process of block representation is applied to $\mathcal{N}$, therefore the feature matrix M_{kpc} is obtained in $O(\mathcal{N}n^2)$ time. After this step, the lexicographical sorting is applied on M_{kpc} that takes $O(\mathcal{N} \log \mathcal{N})$ time. As $\mathcal{N}$ is larger then n or even n^2, therefore the time required against the machine instructions of $O(\mathcal{N}n^2 + \mathcal{N} \log \mathcal{N})$ is $O(\mathcal{N} \log \mathcal{N})$. For comparison of blocks the algorithm takes the linear time $O(\mathcal{N})$ that does not affect the time complexity of the algorithm. Hence, the time complexity of the algorithm remains $O(\mathcal{N} \log \mathcal{N})$, whereas the total machine instructions required are $O(\mathcal{N}(n \log n + n^2) + \mathcal{N} \log \mathcal{N} + \mathcal{N})$.

4. Experimental Results and Discussions

The experimental results of proposed technique are presented in this section. Adobe Photoshop is used to forge the images and all the experiments are performed on a plate form with Intel 1.70 GHz Core i5 processor and MATLAB 2011. The performance of the proposed technique is evaluated on two datasets. We used grayscale images with the size 128 × 128 pixels from the DVMM Columbia University dataset [29]. The second dataset is collected from the Internet, containing the images of sizes 256 × 256 and 512 × 512 pixels. In the experimentation, we set the parameter values for *B (current block) of size* $h \times w = 16 \times 16$, N_n *(number of rows to compare)* = 20, N_t *(block distance threshold)* = 40, N_c *(number of principal components)* = 10 and d_t *(similarity distance between vectors)* = 0.0015, respectively. The experimentation details are presented in the following sections.

4.1. Performance Evaluation. Practically, the most significant property of a detection technique is its capability to discriminate forged and authentic images. In addition to this, the power of locating the forged area correctly is also very important which gives a strong evidence to expose digital forgeries. Thus, the performance of our algorithm is evaluated at two levels: at image level, where we are concerned about the fact that the detected image is truly a forged image, and at pixel level, where we evaluate how accurately the forged areas can be located. To show the accuracy of the proposed technique at image level, the computation of precision "p" indicates the probability that an identified forgery is indeed a forgery; and recall "r" denotes the probability that actually a forged image is detected [25]:

$$\begin{aligned} p &= \frac{T_P}{T_P + F_P}, \\ r &= \frac{T_P}{T_P + F_n}, \end{aligned} \tag{20}$$

where T_P represents the total number of correctly detected forged images, F_P represents the total number of authentic images mistakenly detected as forged, and F_n represents the total number of forged images incorrectly missed.

To show the accuracy at pixel level the true positive rate (TPR) and the false positive rate (FPR) are calculated as follows:

$$\begin{aligned} \text{TPR} &= \frac{\left|\varphi_s \cap \widetilde{\varphi}_s\right| + \left|\varphi_f \cap \widetilde{\varphi}_f\right|}{\left|\varphi_s\right| + \left|\varphi_f\right|} \\ \text{FPR} &= \frac{\left|\widetilde{\varphi}_s - \varphi_s\right| + \left|\widetilde{\varphi}_f - \varphi_f\right|}{\left|\widetilde{\varphi}_s\right| + \left|\widetilde{\varphi}_f\right|}, \end{aligned} \tag{21}$$

where φ_s represents the pixels of original area, φ_f the pixels as the forged area, $\widetilde{\varphi}_s$ the pixels as the detected original area, and $\widetilde{\varphi}_f$ the pixels as the detected forged area. Hence, the TPR shows the performance of technique by correctly identifying the pixels of the copy-moved areas in the forged image, while FPR reflects the pixels which are not contained in forged region but mistakenly included by the implemented technique. Therefore, both the above parameters point out how accurately the proposed technique can locate duplicated areas. The more the TPR is close to 1 and FPR is close to 0, the more precise the technique would be.

4.2. Effectiveness Testing. In order to test the effectiveness of the proposed algorithm, the first experiment for detecting copy-move forgery is performed on the images where the forged region is translated to another location of the image. All the images used in this experiment are without any postprocessing operation. We selected grayscale images from dataset-I with the size of 128 pixels × 128 pixels. The detection

Table 2: Detection results with Gaussian blurring.

	$w = 5, \delta = 0.5$		$w = 5, \delta = 1$	
	24×24	40×40	24×24	40×40
Precision (p)	0.988	0.998	0.980	0.995
Recall (r)	1	1	0.983	1
TPR	0.934	0.955	0.859	0.875
FPR	0.035	0.018	0.054	0.017

Table 3: Detection results with AWGN.

	SNR = 35 dB		SNR = 40 dB	
	24×24	40×40	24×24	40×40
Precision (p)	0.979	0.993	0.980	1
Recall (r)	0.984	0.998	0.991	1
TPR	0.979	0.992	0.985	0.995
FPR	0.055	0.046	0.049	0.027

Table 4: Detection results with JPEG compression.

	$Q = 80$		$Q = 90$	
	24×24	40×40	24×24	40×40
Precision (p)	0.919	0.933	0.970	0.933
Recall (r)	0.925	0.938	0.975	0.981
TPR	0.791	0.909	0.957	0.975
FPR	0.017	0.013	0.013	0.009

results of the experiment can be seen from Figure 3, where left to right is the original, the forged, and the resultant image. As can be seen from Figure 3, there are large similar areas which are difficult to differentiate. However, the algorithm detected the forged regions efficiently. In the second experiment, we selected some color images from dataset-II with the dimensions 512 pixels × 512 pixels. The detection results are given in Figure 4, where left to right is the original, the forged, and the resultant image. We can see from Figure 4 that all the forged objects are irregular; however, the algorithm detected the forged objects precisely.

4.3. Robustness and Accuracy Test of the Algorithm. With the help of image editing software, a counterfeiter usually makes his best efforts to create a forged image. In real life, in order to achieve some purpose, a counterfeiter intentionally performs some postprocessing operations such as blurring, noise, and compression to create imperceptible forged images. Additionally, multiple copy-move forgeries are also a means of image tempering, where there are multiple duplicated areas. In this section, we take these into account and presented some experiments to show the robustness and accuracy of the algorithm. However, in [11, 13, 24, 30], such experiments are not given. The forgery detection results are shown in Figures 5–7. However, the test for detecting multiple copy-move forgeries is given in Figure 8.

Moreover, to evaluate the robustness and accuracy of the proposed technique quantitatively, 100 authentic images are selected from the two datasets for generation of forged images. To obtain four relatively different forged images from a selected authentic image, a square area of size 24×24 pixels is copied from a random location and pasted onto a nonoverlapping area. Adopting the same approach, the squared area of size 40×40 pixels is used to generate four more tempered images for the selected image. In this way the forged image dataset comprising 800 images is generated for the selected authentic images. These forged and original images are then contaminated with postprocessing operations such as Gaussian blurring, AWGN, and JPEG compression. The results are given in Tables 2–4, which evaluate the robustness and accuracy of the algorithm at image and pixel level.

The results given in Tables 2–4 show that the detection performance would be better when the duplicated region is larger. Table 2 indicates that the detection performance of the algorithm is high when the images are blurred by Gaussian blurring; even when the images have poor quality ($w = 5$, $\delta = 1$) and small forged area (24×24 pixels), our technique fails to detect only 14 out of 800 forged images ($r = 0.980$). Table 3 shows that the algorithm also performed well in the case of AWGN distorted images. From Table 3, we can draw a conclusion that the proposed algorithm is also capable of detecting forged regions precisely in the case of slightly compressed images with quality factor ($Q = 80$ and $Q = 90$).

In the last experiment, the proposed technique is compared with other approaches: DCT-based [11], PCA-based [13], FMT-based [12], and improved DCT-based [14]. For this purpose, we selected 100 authentic images of size 512 × 512 pixels and generated 400 forged images. Here, to obtain two relatively different forged images, a square area of size 48×48 pixels is copied from a random location and pasted onto a nonoverlapping area. The overall average performance comparison of over 400 forged images blurred with Gaussian blurring, distorted with AWGN, and with JPEG compression level is shown in Figures 9, 10, and 11, respectively. In the case of Gaussian blurring, Figure 9 indicates the results, where the forged images are blurred by Gaussian filter ($w = 5$ and $\delta = 0.5$, 1, 1.5, 2, 2.5, and 3). It is observed that decreasing the radius of Gaussian filter results in higher TPR but lower FPR for all methods. Moreover, TPR curve of the proposed technique achieves higher performance than other techniques, with TPR ≥ 85%, even when the radius of blurring is increased. The FPR curve also gives satisfactory performance that the proposed technique has lower FPR, even with larger blurring radius ($\delta = 3$). A similar behavior can be observed in the case of noise; the results are shown in Figure 10, where the forged images are distorted with AWGN (SNR = 20, 25, 30, 35, and 40). It is observed for all the techniques that increasing the SNR levels increases the TPR and decreases the FPR. It is also observed that the overall performance of PCA-based method is lower when SNR drops to about 20 dB. However, the proposed technique exhibited better performance by achieving higher TPR and lower FPR than other related techniques. Figure 11 is showing the results with different JPEG compression levels ($Q = 70, 75, 80, 85$, and 90), which indicates that the proposed technique performed well when the forged images were slightly compressed.

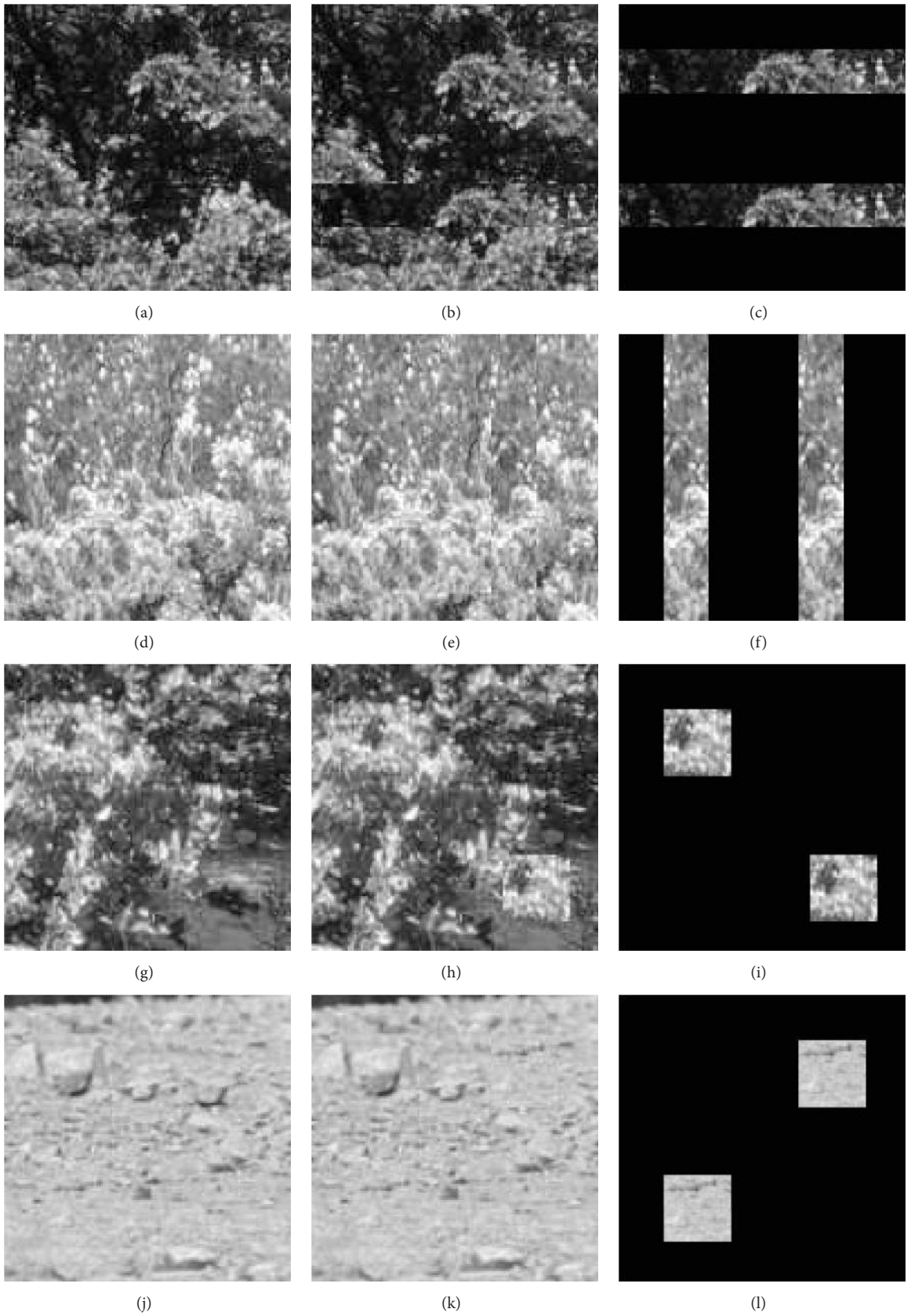

Figure 3: Detection results (top to bottom are the original, forged, and the resultant map image, resp.).

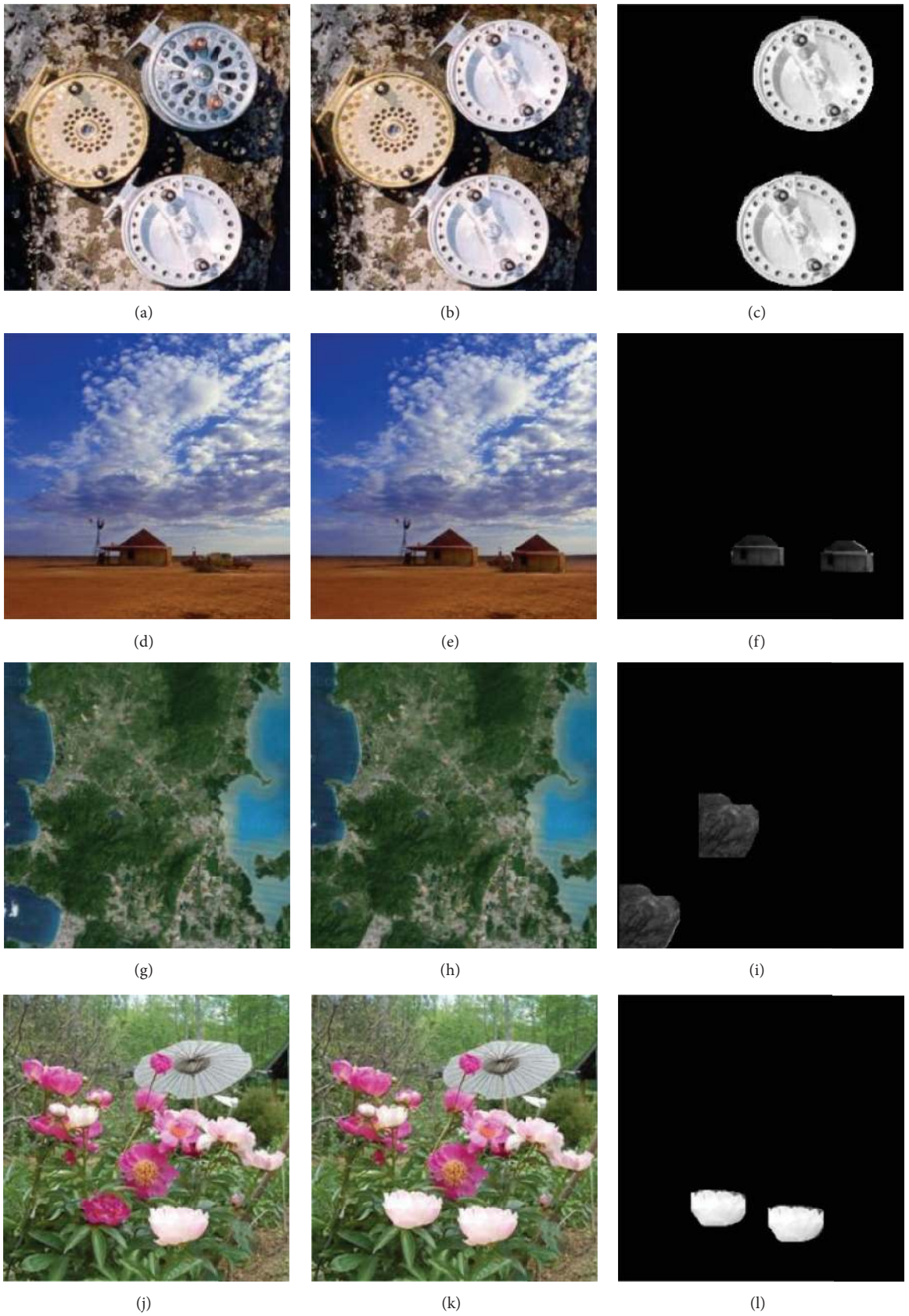

Figure 4: Detection results (top to bottom are the original color image, forged image, and the resultant map image, resp.).

(a) Original image (b) Forged image (c) Output image

Figure 5: Detection results of Gaussian blurring ($w = 5$ and $\delta = 1$).

(a) Original image (b) Forged image (c) Output image

Figure 6: Detection results of AWGN (SNR = 35).

(a) Original image (b) Forged image (c) Output image

Figure 7: Detection results of JPEG compression ($Q = 85$).

5. Conclusion

In this paper, we focused on finding the ways through which we can assure the detection of copy-move forgery in digital images. The main consideration of this paper was to reduce the dimension of the feature length and find the forged objects in the suspected image. Therefore, we have applied DCT and kernel PCA for feature extraction which considers the identical objects found in the forged image. Furthermore, this technique does not require any prior information embedded into the image and works in the absence of digital signature or digital watermark. From the results, a conclusion can be drawn which is that the proposed technique not only effectively detects multiple

(a) Original image (b) Forged image (c) Output image

Figure 8: Multiple copy-move forgeries detection.

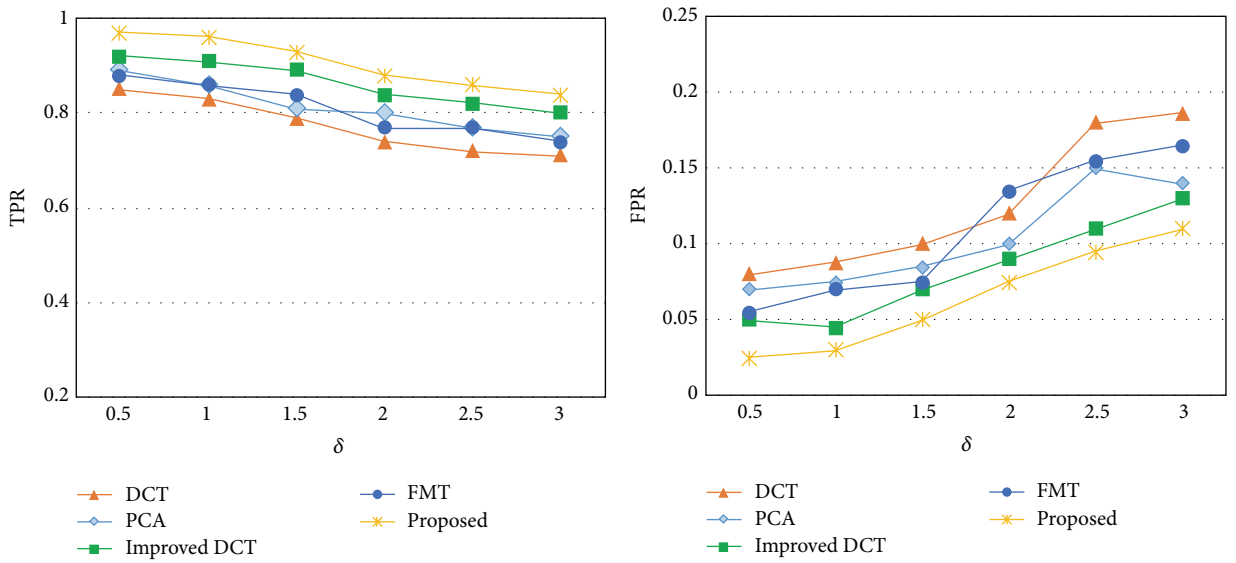

Figure 9: TPR and FPR under different Gaussian blurring.

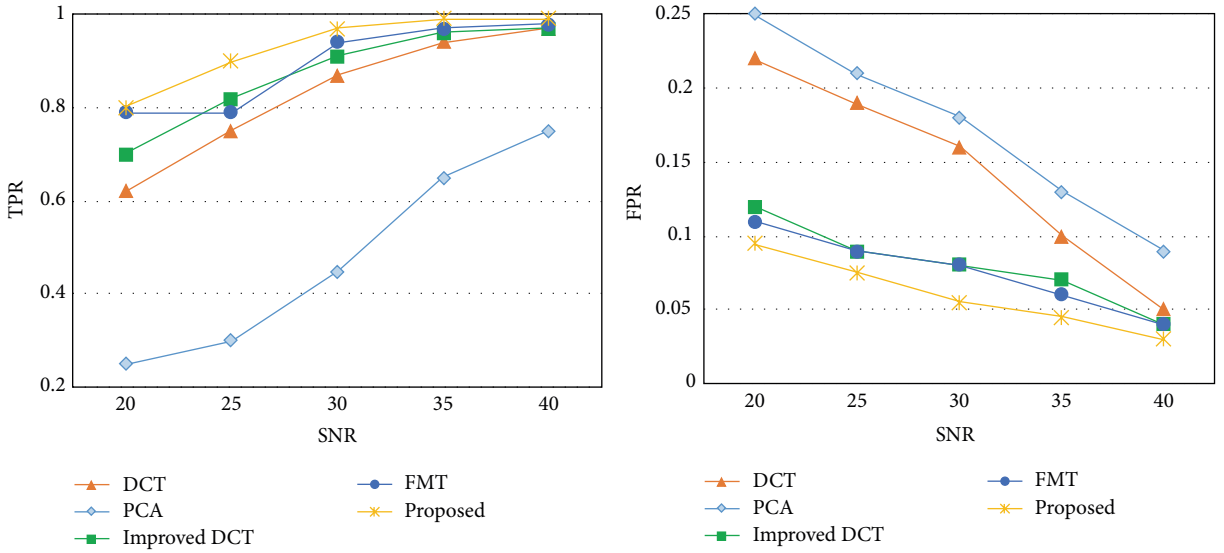

Figure 10: TPR and FPR under different AWGN.

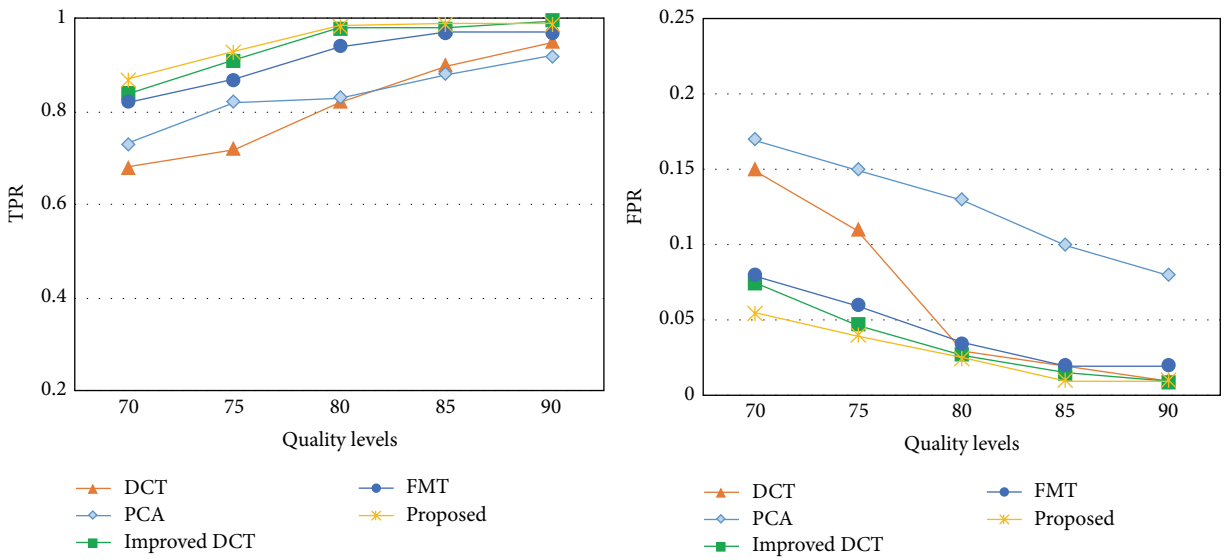

Figure 11: TPR and FPR under different JPEG compression levels.

copy-move forgeries and precisely locates the forged areas but also has nice robustness to postprocessing operations such as Gaussian blurring, AWGN, and compression. Moreover, comparing the detection performance of the proposed technique with existing standard copy-move forgery systems [11–14], the results of our technique are reasonably good in terms of average TPR and FPR.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (2013R1A1A2008180).

References

[1] N. Krawetz, "A pictures worth digital image analysis and forensics," Black Hat Briefings, 2007.

[2] S. Lian and Y. Zhang, "Multimedia forensics for detecting forgeries," in *Handbook of Information and Communication Security*, pp. 809–828, Springer, New York, NY, USA, 2010.

[3] Y. Li, "Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching," *Forensic Science International*, vol. 224, no. 1–3, pp. 59–67, 2013.

[4] H. Farid, "Digital doctoring: how to tell the real from the fake," *Significance*, vol. 3, no. 4, pp. 162–166, 2006.

[5] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing [multimedia authentication technologies]," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 40–49, 2004.

[6] H. Farid, "Image forgery detection: a survey," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.

[7] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, Burlington, Mass, USA, 2007.

[8] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication*, vol. 39, pp. 46–74, 2015.

[9] T. Qazi, K. Hayat, S. U. Khan et al., "Survey on blind image forgery detection," *IET Image Processing*, vol. 7, no. 7, pp. 660–670, 2013.

[10] T. Mahmood, T. Nawaz, R. Ashraf et al., "A survey on block based copy move image forgery detection techniques," in *Proceedings of the International Conference on Emerging Technologies (ICET '15)*, pp. 1–6, Peshawar, Pakistan, December 2015.

[11] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, Cleveland, Ohio, USA, August 2003.

[12] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1053–1056, April 2009.

[13] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Tech. Rep. TR2004-515, Dartmouth College, Hanover, NH, USA, 2004.

[14] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," *Forensic Science International*, vol. 206, no. 1–3, pp. 178–184, 2011.

[15] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 1750–1753, IEEE, Beijing, China, 2007.

[16] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.

[17] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on DCT and SVD," *Forensic Science International*, vol. 233, no. 1–3, pp. 158–166, 2013.

[18] G. Lynch, F. Y. Shih, and H.-Y. M. Liao, "An efficient expanding block algorithm for image copy-move forgery detection," *Information Sciences*, vol. 239, pp. 253–265, 2013.

[19] L. Li, S. Li, H. Zhu, S.-C. Chu, J. F. Roddick, and J.-S. Pan, "An efficient scheme for detecting copy-move forged images by local binary patterns," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 46–56, 2013.

[20] M. Jaberi, G. Bebis, M. Hussain, and G. Muhammad, "Accurate and robust localization of duplicated region in copy-move image forgery," *Machine Vision and Applications*, vol. 25, no. 2, pp. 451–475, 2014.

[21] L. Li, S. Li, H. Zhu, and X. Wu, "Detecting copy-move forgery under affine transforms for image forensics," *Computers and Electrical Engineering*, vol. 40, no. 6, pp. 1951–1962, 2014.

[22] M. Zandi, A. Mahmoudi-Aznaveh, and A. Mansouri, "Adaptive matching for copy-move Forgery detection," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS '14)*, pp. 119–124, Atlanta, Ga, USA, December 2014.

[23] J.-C. Lee, C.-P. Chang, and W.-K. Chen, "Detection of copy-move image forgery using histogram of orientated gradients," *Information Sciences*, vol. 321, pp. 250–262, 2015.

[24] M. Hussain, S. Qasem, G. Bebis, G. Muhammad, H. Aboalsamh, and H. Mathkour, "Evaluation of image forgery detection using multi-scale weber local descriptors," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 4, Article ID 1540016, 2015.

[25] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[26] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring duplicated regions in natural images," *IEEE Transactions on Image Processing*, 2010.

[27] T.-J. Chin and D. Suter, "Incremental Kernel PCA for efficient non-linear feature extraction," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, vol. 3, pp. 939–948, 2006.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 3rd edition, 2009.

[29] Columbia DVMM Research Lab: Columbia Image Splicing Detection Evaluation Dataset, 2004, http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSpliced-DataSet.htm.

[30] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications*, vol. 25, no. 4, pp. 985–995, 2014.