Hindawi
Advances in Multimedia
Volume 2017, Article ID 7130491, 13 pages
https://doi.org/10.1155/2017/7130491

Research Article

Revealing Traces of Image Resampling and Resampling Antiforensics

Anjie Peng,[1,2] Yadong Wu,[1] and Xiangui Kang[2]

[1]*School of Computer Science and Technology, Southwest University of Science and Technology, Sichuan, China*
[2]*Guangdong Key Lab of Information Security, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China*

Correspondence should be addressed to Xiangui Kang; isskxg@mail.sysu.edu.cn

Received 19 August 2016; Revised 23 November 2016; Accepted 12 December 2016; Published 12 January 2017

Academic Editor: Mei-Ling Shyu

Image resampling is a common manipulation in image processing. The forensics of resampling plays an important role in image tampering detection, steganography, and steganalysis. In this paper, we proposed an effective and secure detector, which can simultaneously detect resampling and its forged resampling which is attacked by antiforensic schemes. We find that the interpolation operation used in the resampling and forged resampling makes these two kinds of image show different statistical behaviors from the unaltered images, especially in the high frequency domain. To reveal the traces left by the interpolation, we first apply multidirectional high-pass filters on an image and the residual to create multidirectional differences. Then, the difference is fit into an autoregressive (AR) model. Finally, the AR coefficients and normalized histograms of the difference are extracted as the feature. We assemble the feature extracted from each difference image to construct the comprehensive feature and feed it into support vector machines (SVM) to detect resampling and forged resampling. Experiments on a large image database show that the proposed detector is effective and secure. Compared with the state-of-the-art works, the proposed detector achieved significant improvements in the detection of downsampling or resampling under JPEG compression.

1. Introduction

Resampling is a useful image processing tool, such as upscaling in consumer electronics, downscaling in the online store, social networking, and picture sharing portal. However, some people intentionally utilize the resampling to create tampered images and upload these images to social networks to spread rumors. Due to the abuse in image tampering, resampling forensics attracts researchers' attentions [1–12]. Resampling forensics can also be used to reveal the image's processing history or help people select the secure cover for stenography; for example, Kodovský and Fridrich analyzed how the parameters of downscaling affect the security of stenography [13]. Hou et al. utilized the resampling forensics for blind steganalysis [14]. Therefore, resampling forensics is of particular interest in the multimedia security field.

Early works [1–10] of resampling forensics were based on the periodical artifacts resulting from equidistant sampling and interpolating. These detectors [1–10] can provide reliable results in the uncompressed resampled images. However, their detection accuracies significantly degraded in the case of resampling with JPEG compression. Recent works [11, 12, 15] utilized pattern recognition methods to detect resampling. These works extract the features at first and then perform classification by the machine learning tools. Feng et al. [11] exploited the normalized energy density as the characteristic of image resampling. They divided the DFT frequency spectrum of the second derivative of the image into 19 windows of varying size and then extracted the normalized energy density from each window to form a 19D feature. Li et al. [12] utilized a moment feature to reveal the position and amplitude distribution of resampling in the DFT frequency domain. They first divided the DFT frequency spectrum into 20 subbands with equal interval and then extracted the moment feature from each subband to form a 20D feature. For the sake of simplicity, we called the 19D normalized energy density feature [11] and 20D moment feature [12] as FE and FM, respectively. The machine learning-based detectors [11, 12, 15] get better results than periodical artifacts-based detectors [1–10] for the upsampling with JPEG compression. However, their

performances on the downsampling with JPEG compression still need to be improved. Besides, the above detectors [1–12, 15] have not considered the existence of malicious adversary, a practicable challenge in real life. For instance, Kirchner and Böhme [16] proposed an antiforensic scheme by removing the periodic artifacts with irregular sampling and successfully defeated the periodicity-based approach [1–10]. In the sequel, we called the resampling antiforensics [16] as forged resampling for short.

The appearance of antiforensic technology has been drawing the researchers' attentions to the *security* of the forensics [17, 18]. Sencar and Memon [18] formally define the *security* and *robustness* of the forensics. They pointed out that the *security* concerns the ability to resist intentionally concealed illegitimate postprocessing, while the *robustness* concentrates on the reliability against legitimate postprocessing. In our previous work [19], we employed partial autocorrelation coefficient to reveal the artifacts caused by the forged resampling. Li et al. [15] utilized steganalytic model, SRM, [20] to detect forged resampling and obtained excellent performance.

For a test image, we have no knowledge whether it has been processed by resampling or forged resampling. To avoid missing detection, an alternative approach is that sequentially testing the image by the resampling detector and forged resampling detector. Only if two detectors both predict the image is innocent is the test image taken as an innocent image. To simplify the detection procedure, we propose an integrated detector which can simultaneously detect resampling and its forged resampling. As both the resampled image and forged resampled image are generated via interpolation, we employ the histogram and coefficients of AR model on multidirectional differences to capture the interpolation traces. Experimental results indicate that the proposed integrated detector is effective and secure.

The rest of this paper is organized as follows. Section 2 reviews the resampling forensics and the antiforensic scheme [16]. In Section 3, we introduce a new feature set for resampling forensics. The experiments are presented in Section 4. Section 5 concludes the paper.

2. Background

In this section, we first introduce the resampling and its periodical artifacts and then review the forged resampling scheme proposed by Kirchner and Böhme [16].

2.1. Resampling. The frequently used image resampling operation, including scaling and rotation, consists of two basic operations: (1) resampling, which is also called as spatial transformation of coordinates, and (2) intensity interpolation, which assigns pixel values to the transformed pixels.

Assume that we want to rescale a $r \times c$ image $I(x, y)$ to a $m \times n$ image $E(i, j)$. Generally speaking, 2D image scaling can be separated into two 1D scaling operations along the row and column, respectively. Intuitively, image I is first rescaled along the row to get an intermediate image B of size $m \times c$; then image B is rescaled along the column to get rescaled image E with size of $m \times n$. The whole scaling process can be formulated as

$$E = A^r I A^c = B A^c, \tag{1}$$

where the matrix $A^r(m \times r)$ and $A^c(c \times n)$ determined by the scaling factor s and interpolation kernel embodies the rescaling process for the row and column, respectively. According to (1), we can simplify the discussions of 2D scaling to 1D scaling. As image rotation is similar to image scaling, we concentrate on the image scaling in the following.

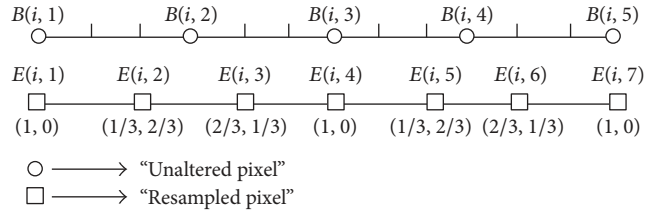


Figure 1: Example of the upscaling ($s = 3/2$, bilinear) process for the ith row of image B (the first line). The corresponding interpolated weights are shown in the bracket.

In the resampling phase, for a scaling factor $s = v/h$ (the greatest common divisor of v and h is 1), the rescaling pixels are first mapped into the original pixel grid with equidistance h/v. Then the intensities of rescaling pixels are calculated by the weighted sums of neighboring original pixel intensities. The weights are determined by the interpolation kernel function, which uses the distances between the rescaled grid and its neighboring original grids as the input.

Due to equidistant sampling, the distance sequences are periodical; thus the interpolated weights are periodical and periodic correlation patterns between neighboring pixels are introduced. Figure 1 shows an example of upscaling ($s = 3/2$) with bilinear interpolation in the ith row of image B. It is shown that the interpolated weights emerge with periodicity equal to 3. In this case, the scaling matrix A^c is as follows:

$$A^c = \begin{bmatrix} 1 & \frac{1}{3} & 0 & 0 & 0 & \cdots \\ 0 & \frac{2}{3} & \frac{2}{3} & 0 & 0 & \cdots \\ 0 & 0 & \frac{1}{3} & 1 & \frac{1}{3} & \cdots \\ 0 & 0 & 0 & 0 & \frac{2}{3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{2}$$

From matrix A^c, we can infer that the $3k$th ($k = 1, 2, 3, \ldots$) column is a linear combination of its 4 neighboring columns, which reveals that the correlations among adjacent pixels are periodical.

Early works [1–10] utilized periodical linear correlations to detect resampling. Popescu and Farid [1] revealed the periodical correlation by a probability map (p-map), which is estimated by the expectation maximum algorithm. For an automatic detector, the periodical artifacts were transformed

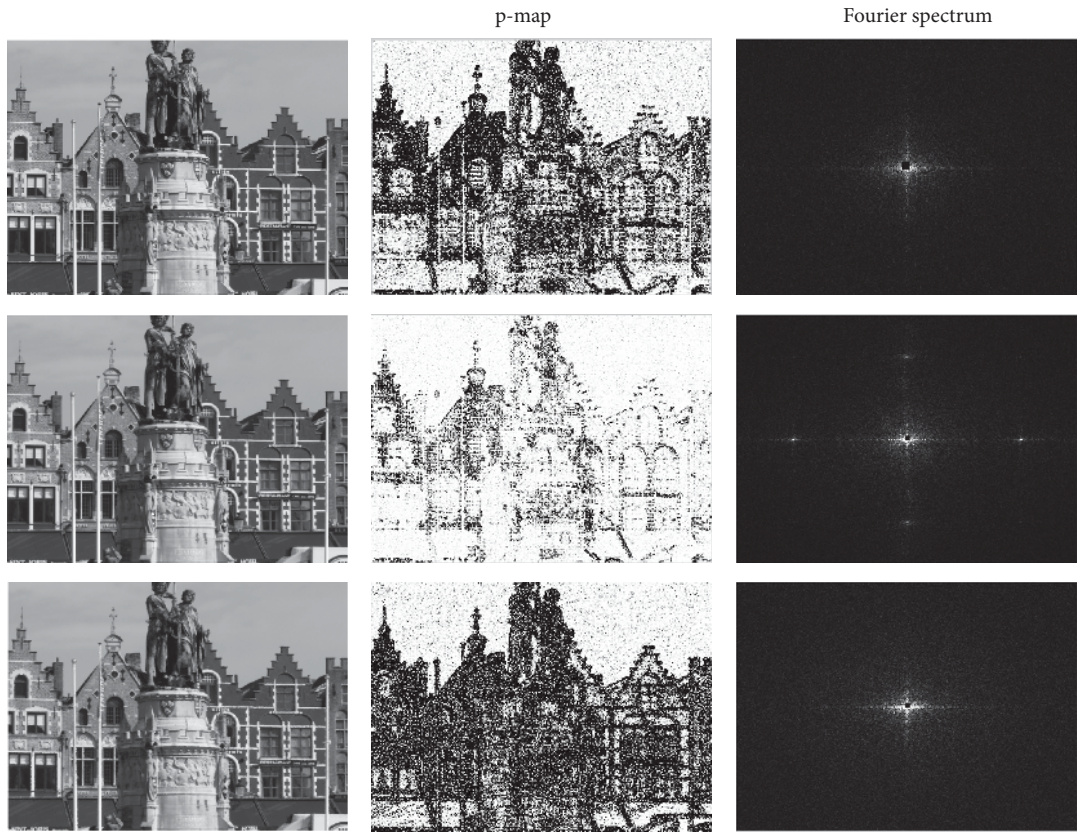


Figure 2: Top row: unaltered image, p-map, and its Fourier spectrum. Second row: upscaled image ($s = 3/2$, bilinear). Third row: forged upscaled image (*attack* 1, $s = 3/2$, bilinear, $\sigma = 0.4$).

as peaks in the frequency domain as shown in the middle row of Figure 2.

2.2. The Forged Resampling Scheme. As the equidistant sampling mainly results in the periodicity appearing in the resampled image, Kirchner and Böhme proposed two attacks to remove that periodicity [16].

(1) The first attack is based on geometric distortion with edge modulation (denoted by *attack* 1). To disturb equidistant sampling, the transformed pixel (i, j) was added by a zero mean Gaussian noise, whose standard variance σ controls the attack strength. That is, the transformed pixel (i, j) turned into a distorted pixel $(i + \varepsilon_1, j + \varepsilon_2)$, where $(\varepsilon_1, \varepsilon_2)$ is the Gaussian noise. Only geometric distortion severely degraded the visual quality, especially at the edge of the image. In order to improve the visual quality, the edge modulation was employed to tune attack strength. Particularly, the attack at the edge was weakened. After unequal sampling, the forged resampled image was obtained by applying the interpolation on the distorted pixel $(i + \varepsilon_1, j + \varepsilon_2)$.

(2) The second attack is dual-path approach (denoted by *attack* 2). This approach applied attacks to the low and high frequency components of the resampled image. In the low frequency path, Kirchner and Böhme applied a nonlinear 5×5 median filter to destroy linear correlations among neighboring pixels. In the high frequency path, they first obtained the residual by subtracting a 5×5 median filtered version from its source image $I(x, y)$ and then applied *attack* 1 on the residual to get the distorted resampled residual. The final forged image is obtained by adding the filtered resampled image and distorted resampled residual.

Both attacks successfully concealed the periodicity in the resampled image; meanwhile they preserved the image's visual quality. Figure 2 demonstrates that an unaltered image (first row) and its forged resampled image (third row) have nearly the same p-map and corresponding Fourier spectrum, which indicates that the periodicity-based detectors [1–10] probably misclassify a forged resampled image as an unaltered image.

3. The Proposed Method

The proposed method aims at classifying the resampled image and forged resampled image from the unaltered image. Such a forensic problem can be formulated as the following hypothesis test:

H_0: the test image is an unaltered image

H_1: the test image is a resampled image or a forged resampled image

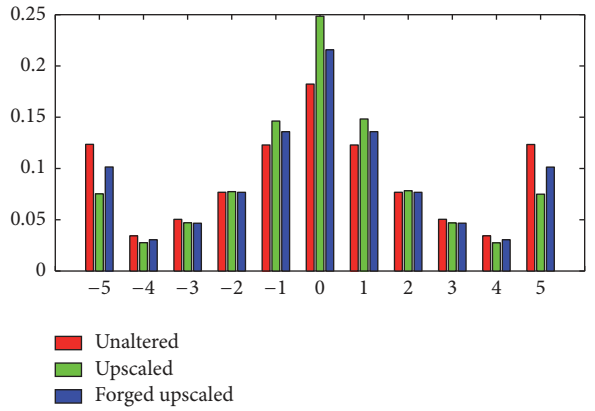


FIGURE 3: The average histogram for 1500 unaltered images, their corresponding upscaled images ($s = 3/2$, bicubic), and forged upscaled images (*attack* 1, $s = 3/2$, $\sigma = 0.4$, bicubic).

3.1. Traces of Interpolation. Compared with unaltered image, due to interpolation, resampled and forged resampled image inevitably leave behind interpolation artifacts. We mainly focus on blurring artifacts and statistical changes of the relationships among neighboring pixels.

In the interpolation phase, the intensity of resampled pixel is a weighted sum of its adjacent original pixels. So, the interpolation is similar to a low-pass filter [11]. The blurring artifact is distinct in the upscaled and forged upscaled images. Figure 3 empirically shows the normalized histograms of the 1st-order horizontal difference ($I(x, y) - I(x, y + 1)$) estimated from BOSSRAW database [21] (please see Section 4 for more details about the database). It can be seen that the upscaled image and forged upscaled image have more bins than the unaltered image in the range of $[-1, 1]$. Because the antialiasing suppresses some high frequency components in the image, the downscaled image with antialiasing also shows slightly blurring artifacts [13]. To capture the blurring artifacts in the resampled and forged resampled image, we will employ the normalized histograms of the image difference as a feature subset.

The interpolation operation also changes the relationships of neighboring pixels. According to formula (1), an unknown scaled pixel is a linear combination of its adjacent l known original pixels, where l is kernel width. For a rescaled image, let us calculate the number of original pixels used to generate p consecutive rescaled pixels. First, the distance of these p rescaled pixels is $(p - 1)/s$, so there are $\lfloor (p - 1)/s \rfloor$ or $\lceil (p - 1)/s \rceil$ original pixels located among these p consecutive pixels, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the floor and ceil function for the nearest integer. Second, the left (or right) $l/2$ original pixels participate in the interpolation process for the starting (or ending) rescaled pixel. *So, we can infer that there are* $\lceil (p - 1)/s \rceil + l$ *or* $\lfloor (p - 1)/s \rfloor + l$ *original pixels used to generate* p *consecutive rescaled pixels.* For example, Figure 1 shows that 6 upscaled pixels $E(i, 1), E(i, 2), \ldots, E(i, 6)$ are generated by 5 original pixels $B(i, 1), B(i, 2), \ldots, B(i, 5)$. According to the above conclusion, we can infer that the relationship of p consecutive rescaled pixels reflects the relationship of $\lceil (p - 1)/s \rceil + l$ or $\lfloor (p - 1)/s \rfloor + l$ consecutive original pixels and of course differs from the relationship of p consecutive original pixels. For the forged rescaled image, due to irregular sampling, some forged pixels cluster closely in the original grid and other pixels located sparsely in the original grid, which also indicates that the forged rescaled image behaves different relationships among neighboring pixels from those of unaltered image. To capture the relationships of neighboring pixels, we will fit the image difference into an AR model and extract AR coefficients as a feature subset. The AR feature can characterize high-order correlations of adjacent pixels, but it has much smaller dimension than SPAM [22] (Subtractive Pixel Adjacency Matrix).

3.2. Multidirectional Differences. As aforementioned, the interpolation is somewhat similar to low-pass filter [11] and thus causes significant statistical changes in the high frequency components of an interpolated image. In general, high frequency components (such as texture and edge) of a natural image are multidirectional. Accordingly, the changes caused by interpolation are also multidirectional. To capture these changes, we first design multidirectional high-pass filters to create multidirectional differences and then extract the proposed feature from these differences.

We employ the kernels of the 1st-order difference to derive multidirectional kernels. The kernels of commonly used 1st-order differences are shown as follows.

The Kernel of the 1st-Order Difference in the Horizontal (H), Vertical (V), Diagonal (D), and Antidiagonal (AD) Direction. It is given as follows:

$$\underset{\text{H}}{\begin{bmatrix} 0, & 0, & 0 \\ 1, & -1, & 0 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{V}}{\begin{bmatrix} 0, & 1, & 0 \\ 0, & -1, & 0 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{D}}{\begin{bmatrix} 1, & 0, & 0 \\ 0, & -1, & 0 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{AD}}{\begin{bmatrix} 0, & 0, & 1 \\ 0, & -1, & 0 \\ 0, & 0, & 0 \end{bmatrix}}. \quad (3)$$

We utilize combinations of horizontal (H), vertical (V), diagonal (D), and antidiagonal (AD) kernels to construct the multidirectional kernels as shown in the following.

Multidirectional Kernel Groups G(1)–G(3). They are given as follows:

$$\underset{\text{G(1)}}{\left[\underset{\text{H}}{\begin{bmatrix} 0, & 0, & 0 \\ 1, & -2, & 1 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{V}}{\begin{bmatrix} 0, & 1, & 0 \\ 0, & -2, & 0 \\ 0, & 1, & 0 \end{bmatrix}} \underset{\text{H + V}}{\begin{bmatrix} 0, & 1, & 0 \\ 1, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{H + V}}{\begin{bmatrix} 0, & 1, & 0 \\ 0, & -2, & 1 \\ 0, & 0, & 0 \end{bmatrix}} \underset{\text{H + V}}{\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 1 \\ 0, & 1, & 0 \end{bmatrix}} \underset{\text{H + V}}{\begin{bmatrix} 0, & 0, & 0 \\ 1, & -2, & 0 \\ 0, & 1, & 0 \end{bmatrix}} \right]}$$

$$
\begin{array}{l}
\text{G(2)} \\
\begin{array}{cccccc}
\text{D} & \text{AD} & \text{D + AD} & \text{D + AD} & \text{D + AD} & \text{D + AD}
\end{array} \\
\begin{bmatrix} 1, & 0, & 0 \\ 0, & -2, & 0 \\ 0, & 0, & 1 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 1 \\ 0, & -2, & 0 \\ 1, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 1, & 0, & 1 \\ 0, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 1, & 0, & 0 \\ 0, & -2, & 0 \\ 1, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 0 \\ 1, & 0, & 1 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 1 \\ 0, & -2, & 0 \\ 0, & 0, & 1 \end{bmatrix} \\
\text{G(3)} \\
\begin{array}{ll}
\text{H + D} & \begin{bmatrix} 1, & 0, & 0 \\ 1, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 1, & -2, & 0 \\ 0, & 0, & 1 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 1 \\ 0, & 0, & 1 \end{bmatrix}
\begin{bmatrix} 1, & 0, & 0 \\ 0, & -2, & 1 \\ 0, & 0, & 0 \end{bmatrix} \\
\text{V + D} & \begin{bmatrix} 1, & 1, & 0 \\ 0, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 1, & 0 \\ 0, & -2, & 0 \\ 0, & 0, & 1 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 0 \\ 0, & 1, & 1 \end{bmatrix}
\begin{bmatrix} 1, & 0, & 0 \\ 0, & -2, & 0 \\ 0, & 1, & 0 \end{bmatrix} \\
\text{H + AD} & \begin{bmatrix} 0, & 0, & 0 \\ 1, & -2, & 0 \\ 1, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 1 \\ 1, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 1 \\ 0, & -2, & 1 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 1 \\ 1, & 0, & 0 \end{bmatrix} \\
\text{V + AD} & \begin{bmatrix} 0, & 1, & 0 \\ 0, & -2, & 0 \\ 1, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 1, & 1 \\ 0, & -2, & 0 \\ 0, & 0, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 1 \\ 0, & -2, & 0 \\ 0, & 1, & 0 \end{bmatrix}
\begin{bmatrix} 0, & 0, & 0 \\ 0, & -2, & 0 \\ 1, & 1, & 0 \end{bmatrix}
\end{array}
\end{array}
\tag{4}
$$

Here "+" means the combination of two directional kernels.

From the 1st-order H or V kernel, the 2nd-order H, V, and H + V kernels (here "+" means the combination of two directional kernel) are derived to reflect the interpolation traces in both H and V directions. Similarly, the 2nd-order D, AD, and D + AD kernels are generated. We also consider the combinations between {H, V} and {D, AD} and obtain four kinds of kernel: H + D, V + D, H + AD, and V + AD. Finally, we have 28 2nd-order filter kernels in total. Following the above way, we can create higher-order kernels. However, the number of higher-order kernels increases sharply, which will increase computation burden in the feature extraction phase, so we only choose aforementioned 28 2nd-order filter kernels to create multidirectional differences.

Based on the kernel's direction, we divide all kernels into 3 groups (denoted by G(1)–G(3) in "*Multidirectional Kernel Groups G(1)–G(3)*"). It is noted that any kernel within a group can be obtained by rotating or flipping other kernels within the same group. We therefore specify that kernels within a group share the same pattern. Considering that spatial statistics in natural images are symmetric with respect to mirroring and flipping [22], we can average the feature sets extracted from the same group to reduce the feature dimension.

To further enhance the interpolation traces left in the high frequencies, besides the image itself, a high frequency spatial residual (denoted by $R^1(x, y)$) is created to construct multidirectional differences. To do this, we firstly divide the Discrete Cosine Transform (DCT) frequency into 3 subbands with equal interval and then select the high frequency subband as shown in Figure 4 to create a high frequency spatial residual $R^1(x, y)$ by the inverse DCT. The whole process can be formulated as

$$
\begin{aligned}
S(u, v) &= \mathrm{DCT}\left(I(x, y)\right), \\
R^1(x, y) &= \mathrm{IDCT}\left(S(u, v) \cdot H(u, v)\right),
\end{aligned}
\tag{5}
$$

where $H(u, v)$ is a high-pass filter. We empirically find that the type of $H(u, v)$ (such as Gaussian high-pass filter) and the partition of the subband have trifling impacts on the resampling detector. For the sake of conciseness, we employ the above proposed method to generate $R^1(x, y)$. For notation convenience, an image $I(x, y)$ is denoted as $R^0(x, y)$ in the sequel.

Each kernel shown in "*Multidirectional Kernel Groups G(1)–G(3)*" is convoluted with an image $R^0(x, y)$ and its high frequency residual $R^1(x, y)$ to generate the 2nd-order difference (denoted by $D(x, y)$). Finally, we get 56 kinds of differences. Inspired by the rich model for steganalysis [20], assembling the feature from the multidirectional differences is expected to be beneficial to the challenging forensic problem, such as detecting resampling in a JPEG compressed image.

3.3. The Feature Construction. In this subsection, we first extract the AR feature (FAR) and histogram feature (FH) from each image difference and then assemble FAR and FH extracted from 56 differences to construct the final feature set.

FAR is extracted based on the direction of $D(x, y)$. (1) For the differences derived from H kernel, as it is mainly used to reflect the variations in the horizontal direction, FAR is extracted in the horizontal direction. (2) Similarly, for the differences created by V kernel, FAR is extracted in

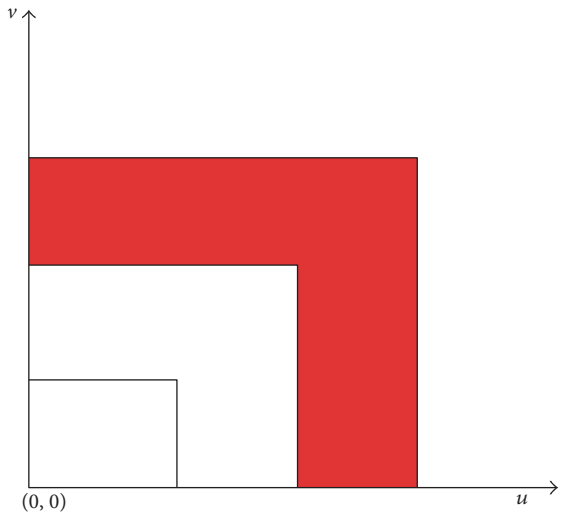


Figure 4: The high frequency DCT subband in red shaded region is used to create $R^1(x, y)$. The coordinate $(0, 0)$ is the DC coefficient.

the vertical direction. (3) For the differences created by other kernels shown in "*Multidirectional Kernel Groups G(1)–G(3)*," the AR coefficients are first extracted from the horizontal and vertical direction, respectively, and FAR are then obtained by averaging them.

Extracting FAR in the horizontal direction is as follows. First, concatenate all rows of $D(x, y)$ to generate a 1D sequence $z = [D(1,:), D^{(\mathrm{LR})}(2,:), D(3,:), D^{(\mathrm{LR})}(4,:), \ldots]$, where $D^{(\mathrm{LR})}(m,:)$ (m is the row index) is a left-right flipped version of the mth row. Then, input z into an AR model formulated to calculate AR coefficients [23]. Transposing $D(x, y)$, we can extract FAR in the vertical direction in the same way. The AR model can be formulated as

$$z(t) = -\sum_{k=1}^{p} a(k)\, z(t-k) + \varepsilon(t), \tag{6}$$

where p, $a(k)$, $\varepsilon(t)$ represent the order, AR coefficients, and prediction error, respectively.

According to the symmetric distribution of the difference as shown in Figure 3, FH is calculated as follows:

$$\mathrm{FH} = \left[h_0, \frac{(h_1 + h_{-1})}{2} \cdots \frac{(h_q + h_{-q})}{2} \right], \tag{7}$$

where h_q (or h_{-q}) is the normalized frequency of the difference element which is equal to q (or $-q$).

To reduce the dimensionality, under the assumption that the kernels within a group belong to the same pattern, we average FAR and FH within the same difference group and denote them as FAR_i^k and FH_i^k (group index $i = 1, 2, 3$; residual index $k = 0, 1$). The proposed feature constructed from multidirectional differences (denoted by FD) is obtained as in (8) by concatenating the feature subset FD^k extracted from $R^k(x, y)$. The dimensions of FAR_i^k and FH_i^k are p and $q + 1$, respectively. Thus, the total dimension of FD is 6 $(p + q + 1)$.

$$\mathrm{FD}^k = \left[\mathrm{FAR}_1^k, \mathrm{FH}_1^k, \mathrm{FAR}_2^k, \mathrm{FH}_2^k, \mathrm{FAR}_3^k, \mathrm{FH}_3^k \right], \quad (k = 0, 1), \tag{8}$$

$$\mathrm{FD} = \left[\mathrm{FD}^0, \mathrm{FD}^1 \right].$$

We set the parameters p and q based on the distributions of AR coefficients and histograms for unaltered images and resampled images. Figure 5 shows the distributions of AR coefficients estimated from BOSSRAW database [21] (please see Section 4 for more details about the database).

For the sake of brevity, we only show the plots for FAR_1^0 and FAR_1^1. Recall that FAR_1^0 and FAR_1^1 are, respectively, extracted from the differences of $R^0(x, y)$ and the differences of $R^1(x, y)$. The subscript "1" represents the fact that the differences are generated by G(1) kernel groups. Both plots show that 12-order AR feature is able to distinguish the scaled or forged scaled image from the unaltered image, so we set $p = 12$. It is shown that FAR_1^0 and FAR_1^1 present different plot shapes, which indicates that they are complementary in the resampling forensics. The parameter q is empirically set as 5, because we observed that most of the difference elements fall within $[-5, 5]$, such as the images in the BOSSRAW database. With $p = 12$ and $q = 5$, the dimension of FD is 108.

The proposed detector is summarized as follows:

(1) Select the high frequency band of DCT as shown in Figure 4 to create the spatial residual $R^1(x, y)$.

(2) Create multidirectional differences by performing the convolution between $R^k(x, y)$ $(k = 0, 1)$ and the kernels in "*Multidirectional Kernel Groups G(1)–G(3)*."

(3) Extract FAR and FH from each difference $D(x, y)$ and construct the proposed feature as (8).

(4) Feed the feature set extracted from the training images into SVM to train the proposed detector.

4. Experimental Results

We test the proposed detector on a composite image database which is comprised of 3000 never resampled images. The BOSSBase [21] and Dresden Image Database (DID) [24] are widely used in the image forensics. Their raw image source database is denoted by BOSSRAW and DIDRAW, respectively. We randomly select 1500 raw images from BOSSRAW and DIDRAW database, respectively, to create the composite database. Before further processing, all images are converted to 8-bit gray images.

The unaltered composite database is provided as the source database for creating resampled image database. We created three kinds of resampled database: upscaling, downscaling, and rotation. We also used the antiforensic method proposed by Kirchner and Böhme [16] to create three kinds of forged resampled database: forged upscaling, forged downscaling, and forged rotation. The commonly used parameters

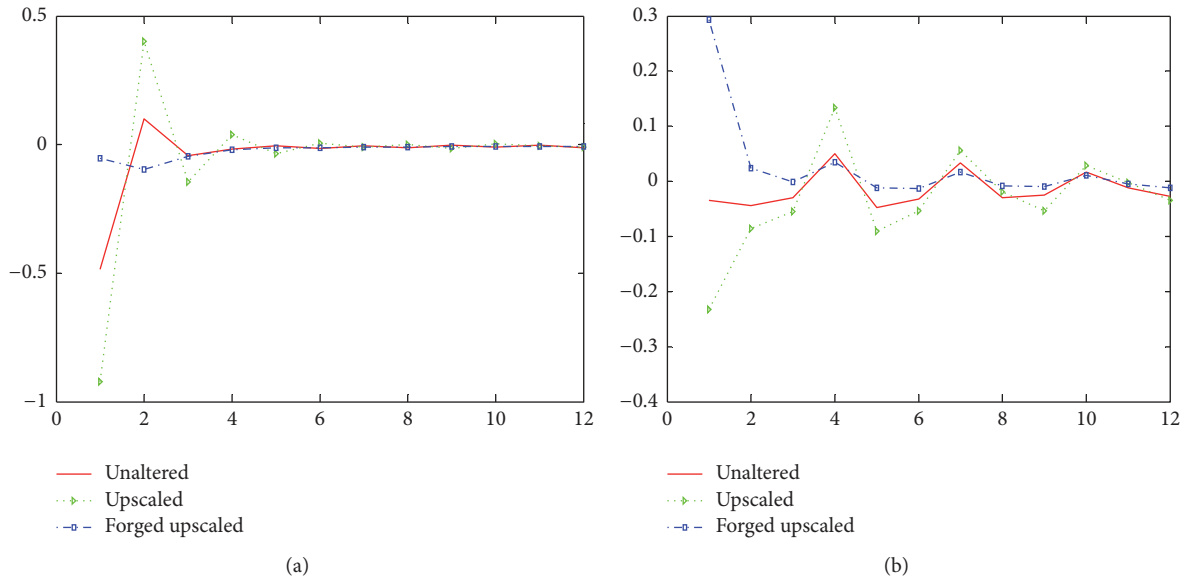


FIGURE 5: The distribution of FAR_1^0 (a) and FAR_1^1 (b) for 1500 uncompressed unaltered images and their upscaled ($s = 3/2$, bicubic) and forged upscaled (*attack* 1, $s = 3/2$, $\sigma = 0.4$, bicubic) versions. X-coordinate: the index of AR coefficient; Y-coordinate: averaged value of AR coefficient.

TABLE 1: Parameters used to create resampled image database.

Database	Parameters
Upscaling (3000 images)	Scaling factors: 1.2, 1.4, 1.6, 1.8 Interpolation kernels: bilinear, bicubic, Lanczos3
Downscaling (3000 images)	Scaling factors: 0.6, 0.7, 0.8, 0.9 Interpolation kernels: bilinear, bicubic, Lanczos3
Rotation (3000 images)	Rotation angle: 5°, 10°, 15°, 20° Interpolation kernels: bilinear, bicubic
Forged upscaling (3000 images)	Scaling factors: 1.2, 1.4, 1.6, 1.8 Interpolation kernels: bilinear, bicubic, Lanczos3 Attack type: *attack* 1, *attack* 2 Attack strength (σ): 0.3, 0.4, 0.5
Forged downscaling (3000 images)	Scaling factors: 0.6, 0.7, 0.8, 0.9 Interpolation kernels: bilinear, bicubic, Lanczos3 Attack type: *attack* 1, *attack* 2 Attack strength (σ): 0.3, 0.4, 0.5
Forged rotation (3000 images)	Rotation angle: 5°, 10°, 15°, 20° Interpolation kernels: bilinear, bicubic Attack type: *attack* 1, *attack* 2 Attack strength (σ): 0.3, 0.4, 0.5

of resampling and forged resampling (depicted in Table 1) are used to generate various types of resampled images. We use the same number for each type in the resampled or forged database. For example, for 12 types of upscaling (four types of scaling factor, three kinds of kernel), we allot each one with 3000/12 = 250 images. To preclude the influence from image resolution, unaltered, resampled, and forged resampled images are center-cropped to 512 × 512.

SVM with Gaussian kernel is employed as the classifier [25]. To avoid overfitting, we conducted a grid-search for the best parameters of SVM by fivefold cross validations on the training set. For training and testing purpose, we created several training-testing pairs. Each pair owns 6000 images, which is comprised by unaltered composite database and its altered version. Training is performed on a random 50% subset of the pair, and testing is performed on the remaining 50%. Hereafter, the same SVM setups are adopted unless particularly specified. The receiver operating characteristic (ROC) curves and the detection error P_e are used to evaluate the SVM-based detector's performance. In formula (9), FPR and TPR denote the false positive rate and true positive rate, respectively.

$$P_e = \min \frac{(\mathrm{FPR} + 1 - \mathrm{TPR})}{2}. \tag{9}$$

To the best of our knowledge, there are no related works which simultaneously detect the resampled image and forged image from the unaltered image. We compare our proposed FD-based detector with the state of the art in the resampling forensics: FE-based detector [11] and FM-based detector [12]. As FE-based detector [11] and FM-based detector [12] have captured some artifacts of interpolation, we suppose they may be effective in forged resampling detection. Additionally, FE detector and FM detector are SVM-based, so it is convenient to compare them with the proposed detector under same experimental settings. We also note that the steganalysis-based detectors [15, 26] have achieved excellent performances in the resampling detection. However, because of huge dimension (34761-D) of steganalysis feature, extracting the 34761-D feature and training the model by the SVM are very time-consuming, so we do not directly compare our method with the steganalysis-based detectors [15, 26].

Table 2: P_e (%) of each detector on detecting resampled images from unaltered images. Here "without" means without applying JPEG compression on test images. The best result is displayed by bold texts.

	JPEG compression	Proposed FD	FM [12]	FE [11]
Upscaled versus unaltered	Without	**0.17**	6.63	13.63
	QF = 95	**1.53**	10.60	14.53
	QF = 80	**11.53**	16.27	18.07
Downscaled versus unaltered	Without	**0.77**	14.87	12.93
	QF = 95	**4.37**	17.90	17.87
	QF = 80	**21.07**	29.50	29.70
Rotated versus unaltered	Without	**0.70**	14.57	23.30
	QF = 95	**3.13**	21.53	23.43
	QF = 80	**13.10**	30.20	28.67

Table 3: P_e (%) of each detector on detecting forged resampled images from unaltered images. Here "without" means without applying JPEG compression on test images. The best result is displayed by bold texts.

	JPEG compression	Proposed FD	FM [12]	FE [11]
Upscaled versus unaltered	Without	**0.13**	5.13	3.30
	QF = 95	**1.40**	8.20	6.90
	QF = 80	**6.10**	18.47	15.47
Downscaled versus unaltered	Without	**0.13**	11.33	16.60
	QF = 95	**3.70**	16.27	20.20
	QF = 80	**16.0**	32.10	30.80
Rotated versus unaltered	Without	**0.20**	8.93	8.13
	QF = 95	**0.90**	12.70	12.30
	QF = 80	**6.67**	26.40	27.47

In the following, we first evaluate the effectiveness of the proposed composite feature. Then, we show that the FD-based detector can not only detect resampled or forged resampled images from unaltered images as traditional method [11, 12, 26] but also simultaneously detect both resampled and forged resampled images from unaltered images. Finally, we give an example of splicing detection using the proposed detector.

4.1. Evaluating Effectiveness of the Composite Feature. The proposed feature FD is a composite of subset FD^k ($k = 0, 1$) as shown in (8). To verify that no subset is redundant, we compared FD with FD^0, FD^1 through detecting upscaled images from unaltered images. As aforementioned, the composite of FD^0 and FD^1 is expected to be beneficial for detecting resampling in a JPEG compressed image. To test FD's robustness against lossy JPEG compression, both unaltered and upscaled images are postcompressed by JPEG 80. With SVM testing, P_e of FD, FD^0, and FD^1 is 7.23%, 8.30%, and 10.57, respectively. This result means that FD yields lowest P_e, which indicates that the feature subset FD^0 extracted from the difference of image and FD^1 extracted from the difference of high frequency residual are collaborative in the resampling classification. In the following, we only reported the result of FD.

4.2. Detecting Unaltered Images from Resampled Images. In this subsection, the proposed detector is tested by distinguishing the unaltered image from the resampled image. To this end, we create 3 uncompressed training-testing pairs: upscaled versus unaltered, downscaled versus unaltered, and rotated versus unaltered and their corresponding JPEG 95 and JPEG 80 version.

Table 2 shows the results for three kinds of feature. Under the uncompressed scenario, the proposed FD-based detector achieves nearly perfect performance ($P_e < 1\%$) for the detections of upscaling, downscaling, and rotation. The FD-based detector performs much better than two other detectors, especially in the detection of downscaling or rotation. For example, in the detection of downscaling without JPEG compression, P_e of the FD-based detector is, respectively, 14.10 percentage points and 12.16 percentage points lower than that of the FM-based detector and FE-based detector. The ROC curves in Figure 6 again verify that the proposed detector has achieved great improvements in the resampling forensics. Under JPEG compression scenario, the FD-based detector also yields lowest P_e in JPEG 90 and JPEG 80 compressed training-testing pairs.

4.3. Detecting Unaltered Images from Forged Resampled Images. In this subsection, we test whether the proposed FD-based detector can resist the malicious attack [16]. The test is designed for distinguishing the unaltered images from the forged resampled images. The FE-based detector [11] and FM-based detector [12] initially do not aim at detecting the antiforensic scheme [16], but they have captured some artifacts of interpolation in the resampled image, such as energy density. Hence, we also test whether FE-based detector and FM-based detector can detect the interpolation artifacts hidden in the forged resampled image.

Table 3 shows the detailed results. Without JPEG compression, the FD-based detector achieves nearly perfect performance ($P_e < 0.2\%$), which indicates that FD-based detector can effectively resist the attacks from antiforensic scheme [16]. Figure 7 shows the corresponding ROC curves under the uncompressed scenario. It can be seen that the ROC curve of the proposed detector is always above that of other two detectors. The advantage of FD-based detector is prominent when FPR is low. For example, in the downscaling detection, the TPR of FD-based detector is 99.93% at FPR = 1%, which is about 59.53 percentage points and 87.63 percentage points higher than that of the FM-based detector and FE-based detector. The FE-based detector and FM-based detector have gotten good performances in the detections of forged upscaling and forged rotation. However, their performances deteriorate in the forged downscaling detections. Under JPEG compression scenario, the results in Table 3 indicate that the proposed FD-based detector also outperforms two other detectors.

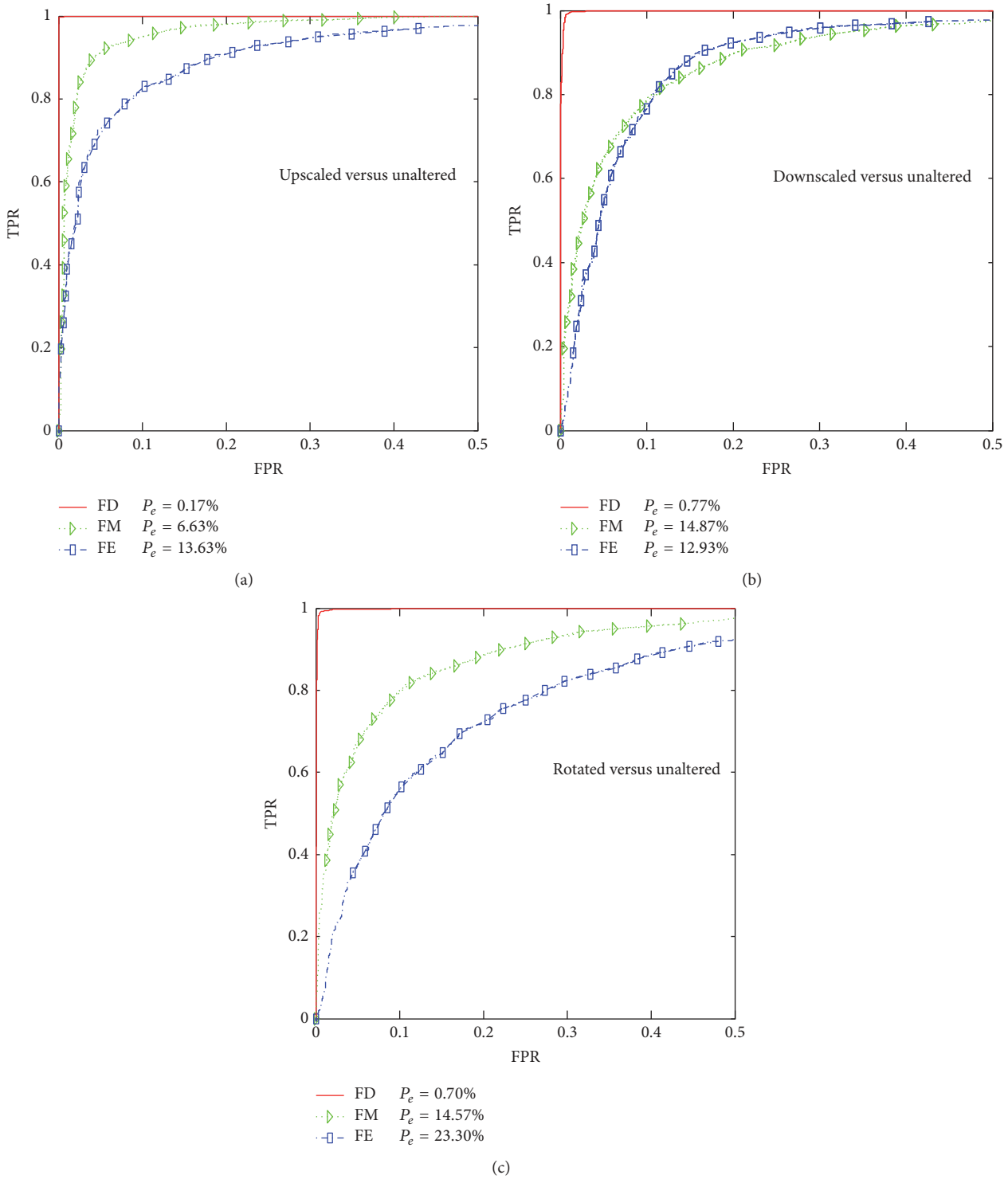


Figure 6: ROC curves showing detections of (a) upscaling, (b) downscaling, and (c) rotation under uncompressed scenario.

4.4. Detecting Unaltered Images from Resampled Images and Forged Resampled Images. In applications, we may have no prior knowledge about the test image. For a more practical detector, we train the SVM detector by unaltered images and "ALL" images including resampled images and forged images. Such a detector requires that the forensic features be distinguishable between heterogeneous images. To visually demonstrate the ability of FD, we map FD into a 2D space by linear discriminate analysis (LDA). Clear distinctions among three types can be seen in Figure 8.

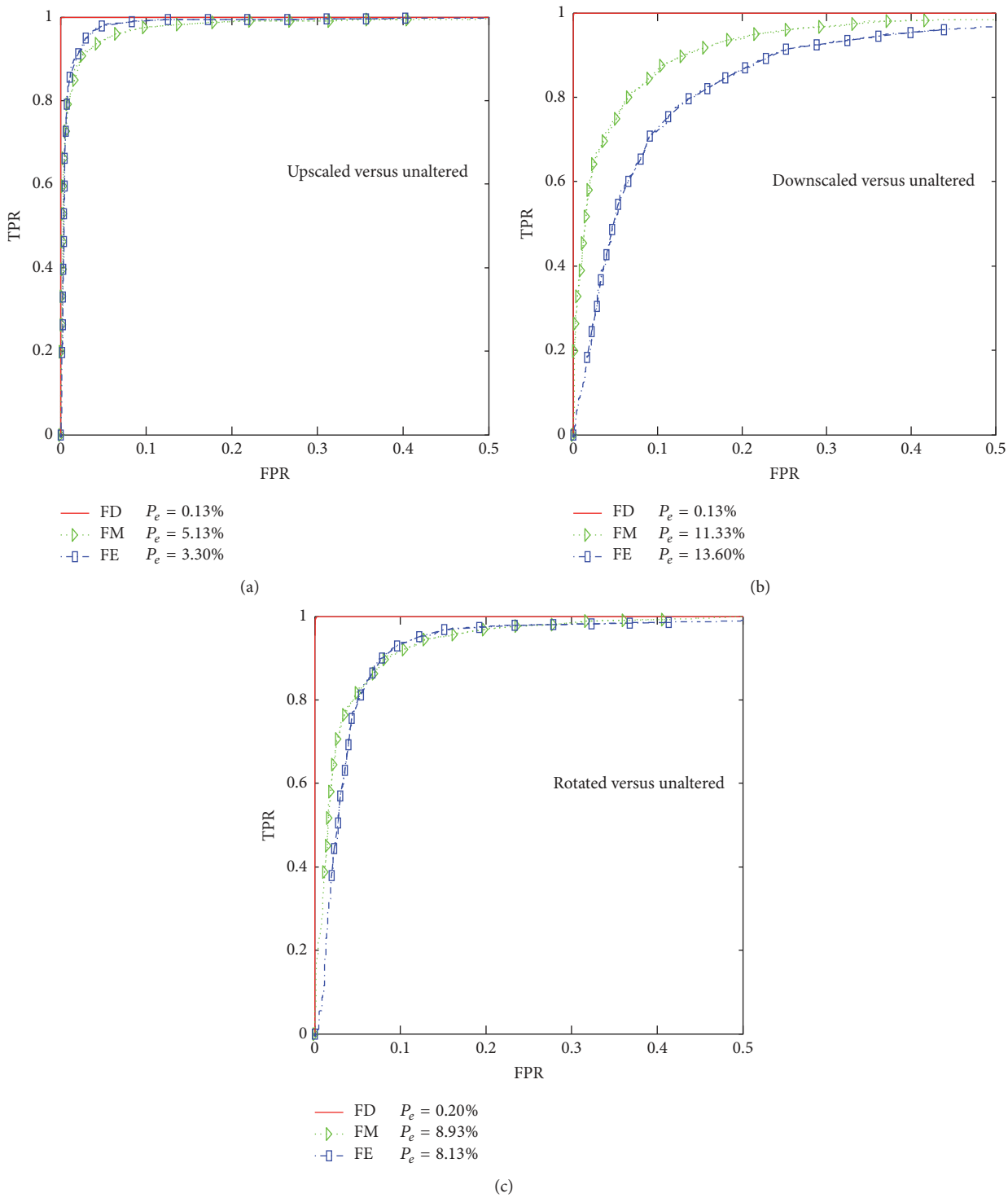


Figure 7: ROC curves showing detections of (a) forged upscaling, (b) forged downscaling, and (c) forged rotation under uncompressed scenario.

We create 3 training-testing pairs in this subsection as shown in Table 4. "ALL" class is comprised by 1500 resampled images and 1500 forged resampled images. We randomly select 500 upscaled images, 500 downscaled images, and 500 rotated images to compose the resampling class in "ALL" database. The forged resampling class is formed by the same manner.

Table 4 gives the detailed results. Under the uncompressed scenario, it can be seen that FD-based detector can effectively distinguish the altered image ("ALL" class)

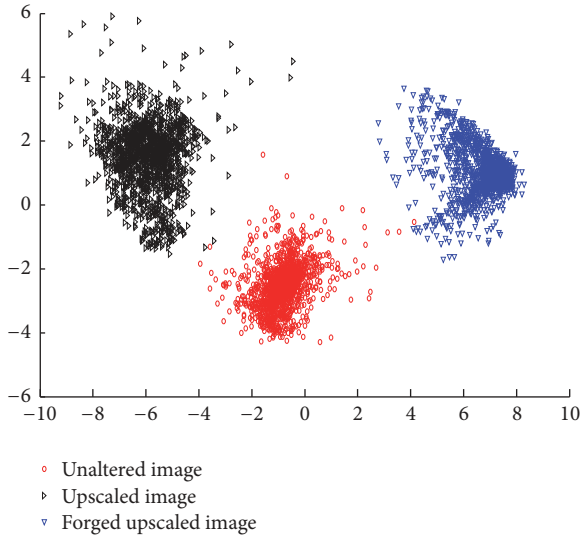


Figure 8: The 2D feature of FD (after LDA) estimated from 1500 uncompressed images of BOSSRAW database.

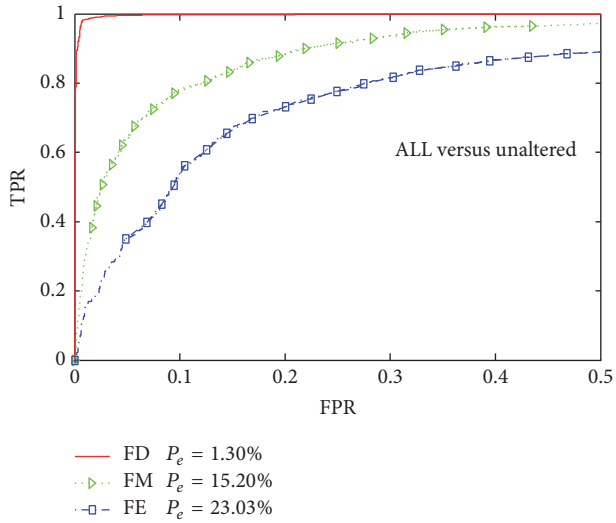


Figure 9: ROC curves of detecting resampled and forged resampled image ("ALL") from unaltered image under uncompressed scenario.

Table 4: P_e(%) of each detector on detecting "ALL" images from unaltered images. Here "without" means without applying JPEG compression on test images. The best result is displayed by bold texts.

	JPEG compression	Proposed FD	FM [12]	FE [11]
ALL versus unaltered	Without	**1.30**	15.20	23.03
	QF = 95	**5.40**	21.90	26.63
	QF = 80	**20.33**	32.73	33.93

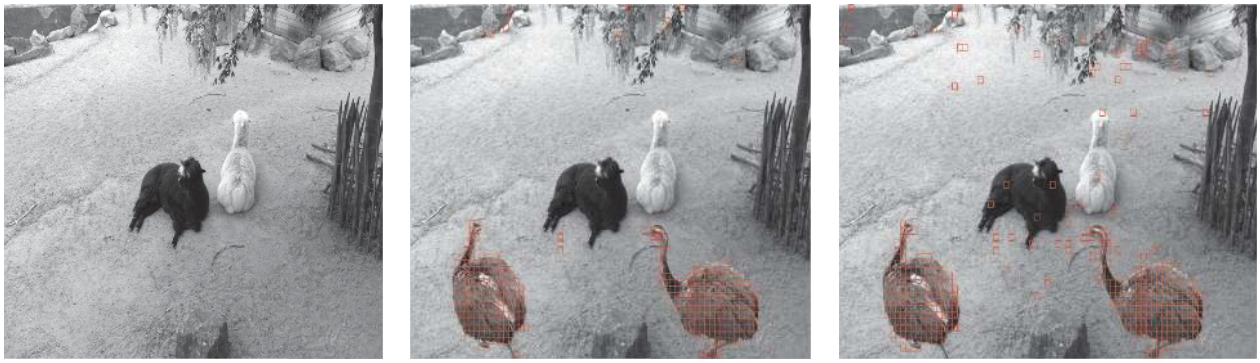
from the unaltered image, which indicates that the proposed feature captures the fingerprints of interpolation which are left in the resampled image and forged resampled image. Either under uncompressed or JPEG compressed scenario, the proposed FD-based detector performs the best. Figure 9 demonstrates that, with FPR = 1%, the FD-based detector achieves TPR = 98.3%, which indicates that the proposed detector is practical in the real applications.

4.5. An Example of Splicing Detection. In this subsection, we use the proposed detector to detect the spliced tampering. Since the location of the pasted object is unknown, the questioned image is divided into nonoverlapped blocks at first, and then each block is predicted by the proposed detector. The block size is set to be 64 × 64. Accordingly, the SVM detector is trained on 64 × 64 blocks. The training set is composed of 3000 unaltered images and 3000 "ALL" images as used in Section 4.3.

Figure 10(b) shows an example of spliced image. It is created by splicing two birds into Figure 10(a). To create convinced tampering, the forger may repeatedly employ the resampling to adjust visual quality. To simulate the real cases, the right bird in Figure 10(b) is first downscaled (scaling factor $s = 0.8$, bicubic) and then upscaled ($s = 1.2$, bicubic). The left bird is processed by antiforensic scheme using default settings [16] (*attack* 2, $\sigma = 0.4$, $s = 0.8$, bilinear). Figures 10(b) and 10(c) show the tampering detection results for the uncompressed and JPEG 95 compressed tampering, respectively. The 64 × 64 block predicted as tampered is marked in red color. Although the proposed detector is only trained on the image blocks with a single scaling operation, it can locate most of the spliced region, including the region which underwent multiple scaling operations. As the edge of the inserted object is a composite of unaltered and altered block, some missing detections emerge in the edge of two pasted birds. Note that this tampered example is simple tampering. In real life, the forgers will adopt various ways to escape the detection of forensic tools. Generalized forensic tools, which can identify usual image operations and their combinations, may be useful in the detections of complicated tampered images.

5. Conclusion

In this paper, we have proposed a novel integrated detector for detecting image resampling and forged resampling, which simultaneously addresses the effectiveness and security concerns. We design multidirectional differences to extract the feature. To capture the traces of resampling and forged resampling, the feature is extracted from the coefficients of the autoregressive model and histograms. Experiments on a large composite image database show that the proposed detector is effective and secure and yields great improvements in the detection of downsampling or resampling under JPEG compression. The tampering detection results illustrate that the proposed detector is promising in practical applications. We have found that the lossy JPEG compression affects the performance of the proposed detector. The performance degrades with increasing JPEG compression ratio. Improving the detector's robustness against heavy JPEG compression is our future work.



(a) (b) (c)

Figure 10: An example showing (a) an unaltered image and tampering detection results for (b) uncompressed and (c) JPEG compressed (QF = 95) tampering. The red box of size 64 × 64 indicates that this box is predicted as tampered by the proposed detector.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was partially supported by NSFC (Grant nos. 61379155, U1135001, and 61303127), the Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20110171110042), NSF of Guangdong province (Grant no. s2013020012788), and Doctoral Research Fund of Southwest University of Science and Technology (Grant no. 16zx7104).

References

[1] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005.

[2] A. C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," in *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV '05)*, pp. 65–72, IEEE, May 2005.

[3] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, 2008.

[4] M. Kirchner, "Fast and reliable rescaling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM Workshop on Multimedia and Security (MM&Sec '08)*, pp. 11–20, Oxford, UK, September 2008.

[5] W. Wei, S. Wang, X. Zhang, and Z. Tang, "Estimation of image rotation angle using interpolation-related spectral signatures with application to blind detection of image forgery," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 507–517, 2010.

[6] M. Kirchner and T. Gloe, "On rescaling detection in re-compressed images," in *Proceedings of the IEEE International Workshop on Information Forensics and Security*, pp. 21–25, 2009.

[7] D. Vázquez-Padín, C. Mosquera, and F. Pérez-González, "Two-dimensional statistical test for the presence of almost cyclostationarity on images," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 1745–1748, IEEE, Hong Kong, September 2010.

[8] D. Vázquez-Padín, C. Mosquera, and F. Pérez-González, "Prefilter design for forensic resampling estimation," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS '11)*, pp. 1–6, Iguacu Falls, Brazil, November 2011.

[9] D. Vázquez-Padín and P. Comesaña, "ML estimation of the resampling factor," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS '12)*, pp. 205–210, IEEE, Tenerife, Spain, December 2012.

[10] D. Vázquez-Padín, P. Comesaña, and F. Pérez-González, "Set-membership identification of resampled signals," in *Proceedings of the 5th IEEE International Workshop on Information Forensics and Security (WIFS '13)*, pp. 150–155, Guangzhou, China, November 2013.

[11] X. Y. Feng, I. J. Cox, and D. Gwenaël, "Normalized energy density-based forensic detection of re-sampled images," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 535–546, 2012.

[12] L. Li, J. Xue, Z. Tian, and N. Zheng, "Moment feature based forensic detection of resampled digital images," in *Proceedings of the 21st ACM International Conference on Multimedia (MM '13)*, pp. 569–572, ACM, Barcelona, Spain, October 2013.

[13] J. Kodovský and J. Fridrich, "Effect of image downsampling on steganographic security," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 752–762, 2014.

[14] X. Hou, T. Zhang, G. Xiong, Y. Zhang, and X. Ping, "Image resampling detection based on texture classification," *Multimedia Tools and Applications*, vol. 72, no. 2, pp. 1681–1708, 2014.

[15] H. D. Li, W. Q. Luo, X. Q. Qiu, and J. W. Huang, "Identification of image operations based on steganalytic features," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[16] M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, 2008.

[17] G. Cao, Y. Zhao, and R. Ni, "Forensic identification of resampling operators: a semi non-intrusive approach," *Forensic Science International*, vol. 216, no. 1–3, pp. 29–36, 2012.

[18] H. T. Sencar and N. Memon, "Digital image forensics," in *Counter-Forensics: Attacking Image Forensics*, Rainer Bohme and Matthias Kirchner, pp. 327–366, Springer, 2013.

[19] A. Peng, H. Zeng, X. Lin, and X. Kang, "Countering anti-forensics of image resampling," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '15)*, pp. 3595–3599, IEEE, Québec, Canada, September 2015.

[20] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[21] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system—the ins and outs of organizing BOSS," in *Proceedings of the 13th Information Hiding Conference*, pp. 59–70, Prague, Czech Republic, 2011.

[22] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[23] S. Kay, *Modern Spectral Estimation*, chapter 7, Prentice-Hall, 1988.

[24] T. Gloe and R. Böhme, "Dresden image database for benchmarking digital image forensics," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 22–26, 2010.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[26] X. Q. Qiu, H. D. Li, W. Q. Luo, and J. W. Huang, "A universal image forensic strategy based on steganalytic model," in *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, pp. 165–170, Salzburg, Austria, June 2014.