Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-7408-8

Image splicing localization using noise distribution characteristic

Depeng Zhang[1] · Xiaofeng Wang[1] · Meng Zhang[1] · Jiaojiao Hu[1]

Received: 6 March 2018 / Revised: 2 January 2019 / Accepted: 22 February 2019
Published online: 13 April 2019

Abstract
Image splicing/compositing is common content tampering operation. In this work, we devote to improve the detection accuracy of the splicing/compositing attack for image, and propose an effective image splicing localization method based on the noise distribution characteristic in image. Firstly, the test image is divided into non-overlapping blocks by using an improved simple linear iterative clustering (SLIC) algorithm. Then block-wise local noise level estimation and noise distribution characteristic estimation are performed to generate distinguishing features. Utilizing the fact that image regions from different sources tend to have larger inter-class difference, the fuzzy c-means clustering is used to identify spliced regions. Compared to existing noise-based image splicing detection methods, experimental results on different datasets have shown that the proposed method has superior performance, especially when the noise difference between the spliced region and the original region is small. Moreover, the proposed method is robust for content-preserving manipulations.

Keywords Image splicing detection · Image splicing localization · Simple linear iterative clustering · Noise distribution characteristic · Fuzzy c-means clustering

1 Introduction

The wide using of powerful image editing software makes it ubiquitous to alter the content of digital images without leaving any visible clues. Although the developments of image editing techniques have provided many post-processing tools for artistic expression in the field of art, the abuse of image editing tools have caused serious problem even lead to crimes. Therefore, it is necessary and urgent to develop reliable and precise image forensics technologies to detection and identification image content change caused by malicious attack, whether from the theoretical research or practical application.

✉ Xiaofeng Wang
xfwang@xaut.edu.cn; xfwang66@sina.com.cn

[1] Xi'an University of Technology, Xi'an 710048 Shaanxi, China

Digital image passive forensics has been an active research area during the last decade. It is based on the assumptions that different imaging devices or imaging processing would introduce different inherent patterns into the outputted images. These underlying patterns are consistent in the original images and would be altered after some image manipulations. Thus, it can be used as evidence for image source identification and forging detection [19]. Image forging means that the image content is altered by malicious attacks. Image splicing is a widespread forging operation, and it is done by copying a part of image content and pasting to another image to create a composited image. Since human's eyes can be cheated by spliced images even without any post-processing, so image splicing detection has attracted great attention in recent years. Figure 1 presents two examples of splicing image. In Fig. 1, (a) "Tibetan antelope" is one of the China's top ten most influential news photos in 2006 (come from China central television (CCTV)). In fact, it is a composograph created by splicing two independent images, train and Tibetan antelope. (b) Hidden heaven of the earth, it is a documentary named Patagonia, broadcasted by BBC. It is reported that the volcanic eruption showed in the video system was created by splicing two video pieces via computer. To obtain real visual effect, lightning was combined with erupted ash cloud, and created splendid volcanic thunderstorms.

Since different image have different intrinsic characteristics, while splicing forgery will result in intrinsic inconsistency in forged image. This kind of intrinsic inconsistency can be considered as the forensics feature to detect image splicing attacks. To capture such intrinsic inconsistency, many methods have been proposed during the last decade.

1.1 Related works

In the last decades, a lot of researchers have devoted to the study of image splicing detection. The existing techniques in literature include two main categories: image splicing detection and splicing region localization. The former is a deterministic method that only detects whether an image has undergone splicing operations. The latter is a computational method that can find the position and shape of spliced image regions.

The earlier technologies were mainly focusing on image splicing detection. This kind of methods cast the problem as image classification task. It usually selects well designed image features, and fed into a trained classifier to distinguish spliced images from authentic images. The typical method is the statistical features-based methods. This kind of methods is based on

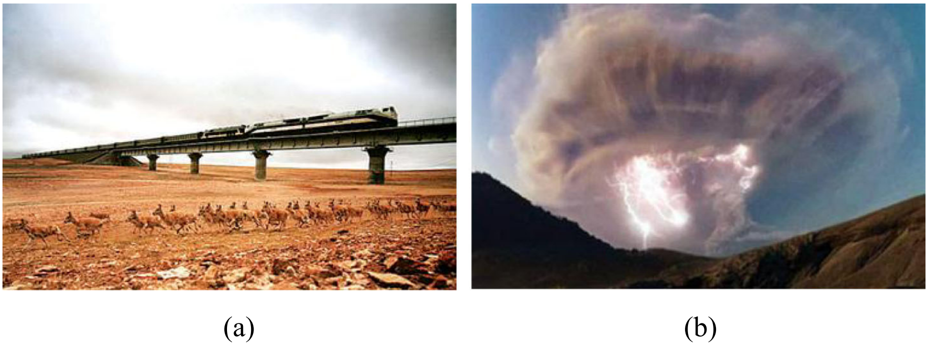


Fig. 1 The examples of image splicing. **a** Tibetan antelope, it is created by splicing train and Tibetan antelope (**b**) Hidden heaven of the earth, lightning was combined with erupted ash cloud

the fact that image splicing might cause the change of inherent statistical features. E.g, [33] and [44] used image chroma component as distinguishing features and SVM as a classifier to detect image splicing. Work [17] suggested a method that extract neighboring joint density features of the DCT coefficients, and use SVM as classifier to detect image splicing. Since image splicing might destroys the texture correlation in the edge of the spliced regions, so the inherent inconsistencies of texture features can be used to detect image splicing. For example, work [32] used gray level co-occurrence matrix (GLCM) to construct features; works [1, 2, 42] used Local binary patterns (LBP) [21] as features to detect image splicing; work [36] reported a method to detect changes caused by seam modification on JPEG retargeted images.

From the view of tampering detection, the real challenge is splicing region localization. With the publishing of the Columbia Image Splicing Detection Evaluation Dataset [7], a lot of scholars were devoted to the study of splicing region detection. Recent years, many splicing region detection methods were proposed.

Johnson and Farid [14] developed a method to identify splicing image regions by detecting the inconsistency in lighting with respect to different parts in an image. Consider that the camera response function (CRF) is a fundamental property of the cameras, and it maps input irradiance to output image intensity, some CRF-based methods [11, 12] are proposed to detect spliced image regions. Fang et al. [39] detect spliced regions by evaluating consistency check of camera characteristics among different areas in an image. Zhang et al. [41] introduced a method based on the planar homography constraint to locate the fake regions roughly and an automatic extraction method using graph cut with online feature/parameter selection to segment the faked objects. Recent work [13] reported a method for detecting image forgery by locating grid alignment abnormalities in JPEG compressed image. Work [34] used a SVM for classing to detect gamma transformation in image and then used it to detect the tampered image regions. Work [29] proposed a model that is based on features extracted from DCT coefficients and Multi-Scale LBP, and presented a method to detect image splicing. Other methods such as [40] and [18] detect image composites by estimating the shadows in image.

Consider that image splicing may influence the features of blur in images, and thus, blur can be used as a cue for image splicing detection. Some defocus blur based splicing detection methods were proposed, for example, [3, 5], and [30]. Recently work [4] reported an image splicing localization method that is based on the partial blur type inconsistency.

Since image splicing may destroy the correlation introduced by the Color Filter Array (CFA) interpolation process, many works are devoting to detect image splicing/ composite by identifying the inconsistency of CFA interpolation pattern. The typical method was proposed in [23]. Gallagher and Chen developed a method [9] that detects image tampering by measuring the presence of demosaicing in digital image. Wang et al. [31] introduced a technique to locate image splicing via re-demosaicing.

Due to the limitation of manufacturing technique and the characteristic of material, imaging devices have some internal defects inevitably. These defects will present in the outputted image in the form of special pixels or pattern noise. In a spliced/composited image, spliced regions that are come from different origins may have different noise characteristics, and this inconsistency can be regarded as an evidence to identify spliced regions.

Lyn e.t. [20] reported a method to expose region splicing by revealing inconsistencies in local noise levels. This method was based on a blind noise estimation algorithm. Pun e.t.[24] proposed an image splicing detection method based on multi-scale noise estimation. In their work, the test image was segmented into superpixels of multiple scales. Those segments not constrained by the noise level function of individual scale were regarded as suspicious regions.

Consider that the principal component analysis (PCA) is a common method of noise estimation, work [38] proposed a method by using PCA-based noise level estimation. In their work, authors perform block-wise noise level estimation of a test image with PCA, and segment the tampered region from the original region by k-means clustering.

Since sensor pattern noise can be considered as the camera's fingerprint, work [6] reported an approach to detect image forgeries using sensor pattern noise. In their work, authors casted the problem in terms of Bayesian estimation, and modern convex optimization techniques were then adopted to achieve a globally optimal solution and photo-response non-uniformity (PRNU) noise estimation. Consider that noise in image forensic applications is generally weak and content-related, work [37] described a noise level function (NLF) that better fits the actual noise characteristics. By exploring the relationship between NLF and the camera response function (CRF), authors formulated a Bayesian maximum a posteriori (MAP) framework to optimize the NLF estimation, and developed a method for image splicing detection according to noise level inconsistency in regions taking from different images.

In recent years, with the widely application of neural network model in pattern recognition field, learn-based detection methods are presented successively, e.g. [8]. These methods are effective and accurate to detect whether an image has undergone splicing operations. Work [27] presents a technique that utilizes a fully convolutional network (FCN) to localize image splicing attacks. Many splicing detection schemes based on Markov features in transform domain have been proposed [10, 43] et al. Work [35] using the difference between the edges of the spliced area and the edges of the natural image, detection of tampered images will become more accurately. Experimental results show that their method outperform some existing splicing localization algorithms.

1.2 Our contributions

Although many image splicing localization methods have been proposed in recent years, there are still some challenging problems, such as the accuracy and robustness of the algorithm. In this work, we devote to improve the detection accuracy and the robustness for content-preserving manipulation, and propose an image splicing/compositing detection technique. Firstly, the test image is divided into non-overlapping blocks by using the Simple linear iterative clustering (SLIC). Then we perform a block-wise local noise estimation in the questioned image, and use the fuzzy c-means clustering to identify spliced image regions. Compared to the existing noise-based image splicing region detection methods, the proposed method has higher detection accuracy and robustness, especially when the noise difference between the spliced region and the original region is small.

The rest of this paper is organized as follows. The proposed image splicing/compositing detection method is described in Section 2. In Section 3, we present experimental results and performances analysis. Section 4 concludes the paper with some thoughts on future works.

2 The proposed method

The proposed method consists of four pipeline stages: image segmentation, local noise level estimation, noise distribution characteristic estimation, splicing region localization.

2.1 Improved SLIC-based image segmentation

Simple linear iterative clustering (SLIC) is a simple and efficient algorithm to construct superpixels. Since this algorithm can generate compact and nearly uniform superpixels, and has a higher comprehensive evaluation in terms of computing speed, object contour preservation, superpixel shape, etc., so it meets the expectation segmentation effect of our work. Moreover, the algorithm groups the pixels by using the similarities among the features of pixels and replaces multitude pixels with a few superpixels to express the image features, which greatly reduces the complexity of image post-processing. Therefore, in this section, we used SLIC as the pre-processing of the proposed algorithm.

However, for traditional SLIC algorithm, the numbers of superpixels depend on artificially setting. In our work, we improve the traditional SLIC algorithm, and propose a method that uses the size of image to self-adaptively determine the number of superpixels. Our approach overcomes the randomness and uncertainty caused by artificial observation and experience in determining the number of image partitions. In particular, it is very meaningful for the batch processing image of different sizes.

In order to establish an approach that can self-adaptively determine the number of superpixels, we take 200 same size images randomly from the image database. We process these images by adding zero mean Gaussian noise with variance $\sigma = 5$, then segment these images uniformly as non-overlapping blocks with size $n \times n$, $n = 16, 32, 64, 128$, respectively, and the schematic diagram is shown in Fig. 2 (a), (b), (c), and (d). We estimate the noise level of each block by using PCA, respectively, calculate noise mean and variance of all blocks, and the results are listed in Table 1.

As can be seen from the Table 1, when $n = 64$, the estimated noise mean is 4.588, it is the closest to 5. This means that the noise estimate is relatively accurate when $n = 64$. Therefore, it is reasonable to use $n = 64$ as the segmentation size.

For $M \times N$ test image I, let $Q = M \times N$. We define the number of superpixels as:

$$K = \begin{cases} 200, & k \geq 150 \\ k, & 100 \leq k \leq 150 \\ 100, & others \end{cases} \tag{1}$$

Here $k = \frac{Q}{n \times n}$, $n = 64$.

We use SLIC algorithm to segment the image I into certain superpixel blocks (seeing Fig. 3 (a)), denote the segmented image as I' (seeing Fig. 3(b)), denote each superpixel block as $S_i(i = 1,2,...,K)$, where K is the number of superpixels.

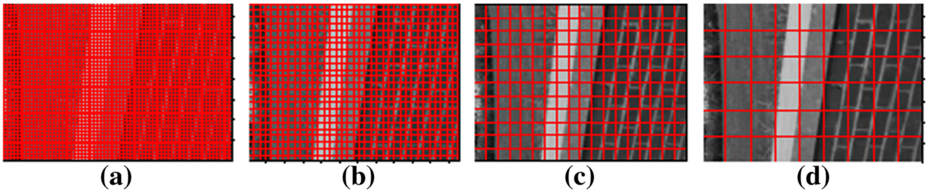


Fig. 2 Different size segmentation diagram. These images uniformly as non-overlapping blocks with size, $n =$ 16, 32, 64, 128, respectively

Table 1 The comparison of noise means and variances for different blocks (The ground truth $\sigma = 5$)

n	16	32	64	128
Noise mean	3.1707	4.4905	4.5880	3.856
Noise variance	0.3371	0.4258	0.3981	0.3357

2.2 Noise standard deviation estimation

2.2.1 Blocks filled and superpixel matrix

To use PCA to estimate the noise variance of each superpixel block, it is necessary to transform the irregularly superpixel block into regular one. To this end, we fill each $S_i(i = 1,2,...,K)$ into a rectangular area according to the horizontal maximum value and vertical maximum value, respectively, and the pixels of filled part come from the surrounding area of the superpixel block in the original image. As shown in Fig. 3(c), the red area indicates an irregular superpixel block; we want to fill it into a regular area, seeing the green rectangle. In Fig. 3(d), black spots indicate the pixels included in the superpixel block, and white spots indicate the pixels that need to be filled. In Fig. 3(e), red spots indicate filled pixels that come from the area inside the green rectangle and outside the superpixel block. Without loss of generality, in $M \times N$ matrix I_{MN}, let red frame represents a superpixel block, green frame represents corresponding filled regular black.

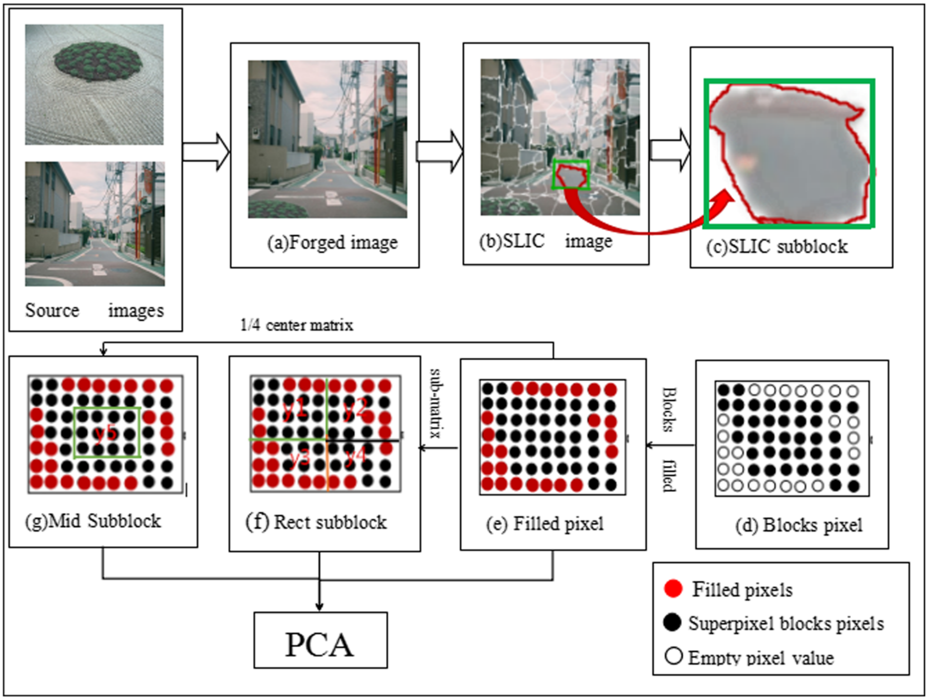
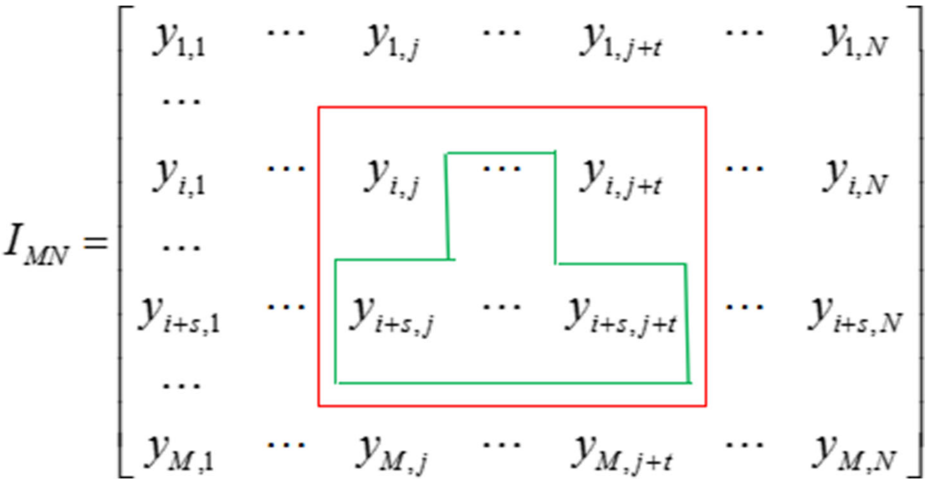


Fig. 3 The framework of PCA-based noise standard deviation estimation


$I_{MN} =$
$y_{1,1}$ ⋯ $y_{1,j}$ ⋯ $y_{1,j+t}$ ⋯ $y_{1,N}$
⋯
$y_{i,1}$ ⋯ $y_{i,j}$ ⋯ $y_{i,j+t}$ ⋯ $y_{i,N}$
⋯
$y_{i+s,1}$ ⋯ $y_{i+s,j}$ ⋯ $y_{i+s,j+t}$ ⋯ $y_{i+s,N}$
⋯
$y_{M,1}$ ⋯ $y_{M,j}$ ⋯ $y_{M,j+t}$ ⋯ $y_{M,N}$

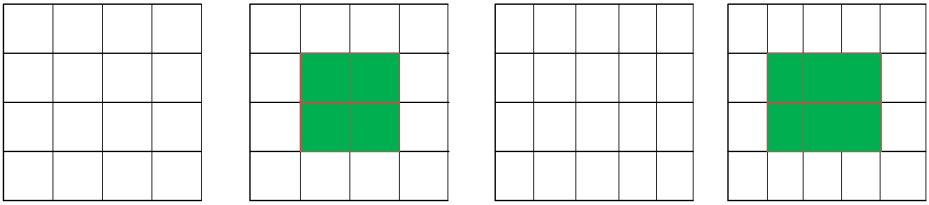
Let the matrix of each filled superpixel blocks S_i be denoted as $R_{i0}(i = 1, 2, \ldots, K)$.

$$R_{i0} = \begin{bmatrix} y_{i,j} & \cdots & y_{i,j+t} \\ \vdots & \ddots & \vdots \\ y_{i+s,j} & \cdots & y_{i+s,j+t} \end{bmatrix}$$

2.2.2 The noise standard deviation

To extract noise feature, the matrix $R_{i0}(i = 1, 2, \ldots, K)$ is divided into four sub-matrix R_{i1}, R_{i2}, R_{i3}, R_{i4}. As shown in Fig. 3(f). If the row or column of R_{i0} is even, divide row or column equally (seeing the Fig. 4 (a)); if the row or column is odd, the pixels in the middle row or column are divided into the next sub-matrix (seeing the Fig. 4 (b)). Next, take 1/4 center matrix of R_{i0} as a new matrix R_{i5} (as shown in Fig. 3 (g)).

Let $B_{i,X}$ denote the sample covariance matrix of $R_{i,X}$, let $\lambda_{i,X}^{(1)} \geq \lambda_{i,X}^{(2)} \geq \cdots \geq \lambda_{i,X}^{(M)}$ (Descending order) be the eigenvalues of $B_{i,X}$ with the corresponding normalized eigenvectors $V_{i,X}^{(1)}, V_{i,X}^{(2)}, \cdots, V_{i,X}^{(M)}$, where, $i = 1, 2, \cdots, K$, $X = 0, 1, 2, \cdots, 5$. According to work [26], $\left(V_{i,X}^{(1)}\right)^T R_{i,X}, \left(V_{i,X}^{(2)}\right)^T R_{i,X}, \cdots \left(V_{i,X}^{(M)}\right)^T R_{i,X}$ are the sample principal components of $R_{i,X}$, and have the following characteristic:



(a) Case 1

(b) Case 2

Fig. 4 **a** divides row or column equally. **b** the pixels in the middle row or column are divided into the next sub-matrix

$$s^2\left(\left(V_{i,X}^{(b)}\right)^T R_{i,X}\right) = \lambda_{i,X}^{(b)}, b = 1, 2, \cdots, M \tag{2}$$

Where $s^2(\cdot)$ represents the variance of $\left(V_{i,X}^{(b)}\right)^T R_{i,X}$, M represents the number of eigenvalues and eigenvectors of the corresponding covariance matrix.

Therefore, the following equation can be used to obtain the corresponding block noise standard deviation $\sigma_{i,\ X}$ please refer to [26] for details:

$$\lim_{N\to\infty} E\left(|\lambda_{i,X}^{(b)} - \sigma_{i,X}^2|\right) = 0 \tag{3}$$

Equation (3) represents $\lambda_{i,X}^{(b)}$ convergence in mean to $\sigma_{i,X}^2$. Therefore, the noise variance can be estimated as $\lambda_{i,X}^{(b)}$, Since convergence in mean implies convergence in probability, $\lambda_{i,X}^{(b)}$ is a consistent estimator of the noise variance. Therefore, we have

$$\sigma_{i,X} = \sqrt{\lambda_{i,X}^{(b)}} \tag{4}$$

Using formula (4), we can get the noise standard deviations $\sigma_{i,\ X}$, ($X = 0, 1, 2, \cdots, 5$), and the average noise standard deviation $\overline{\sigma}_i$:

$$\overline{\sigma}_i = \frac{1}{6}\sum_{X=0}^{5}\sigma_{i,X} \tag{5}$$

Here, $X = 0,1,...,5$. Repeat above steps for all blocks S_i to get $\overline{\sigma}_i(i = 1, 2, \cdots, K)$.

2.3 The estimation of noise distribution characteristic

Work [24, 25, 28] and [15] reported, in certain brightness, the image noise level conforms to the Poisson distribution. To describe the characteristics of noise distribution, we investigate the relationship between noise distribution and pixel intensity by using the method of work [24].

(1) Let $I_W^{'}$ represent the Weiner filtered image of I',

$$I_W^{'} = W\left(I^{'}\right) \tag{6}$$

Here, $W(\bullet)$ represent the Weiner filter. The residual image E_{MN} is as follows:

$$E_{MN} = I^{'} - I_W^{'} \tag{7}$$

Let $E_{MN}(S_i)$ represent the residua of superpixel block S_i. Computing the standard deviation $\sigma(S_i)$ of $E_{MN}(S_i)$, $i = 1, 2, \cdots, K$.

$$\bar{E}(S_i) = \frac{\sum_{j=1}^{|S_i|} E(P_j)}{|S_i|} \tag{8}$$

$$\sigma(S_i) = \sqrt{\frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\left(E(P_j) - E(\bar{S}_i)\right)^2} \tag{9}$$

Where, $|s_i|$ is the number of pixels include in superpixel block S_i, $E(P_j)(j = 1, 2, \cdots, |S_i|)$ represents the intensity value of j-th pixel in $E_{MN}(S_i)$,$E(\overline{S}_i)$denotes the mean value of pixel in $E_{MN}(S_i)$, $\sigma(S_i)$ indicates the standard deviation of pixel in $E_{MN}(S_i)$, it is also the noise standard deviation of S_i.

Computing the mean value $b(S_i)$ of the pixel intensity in $S_i(i = 1, 2, \cdots, K)$.

$$b(S_i) = \frac{\sum_j p_i(j)}{|s_i|} \tag{10}$$

Where $p_i(j)$ is pixel intensities, $|s_i|$ is the number of pixels include in the superpixel block S_i, $b(S_i)$ denotes the mean pixel intensity of the $i-$th superpixel block S_i.

(2) According to [24], the least squares polynomial curve fitting technique [16] can be used to construct noise level function, and five-degree curve fitting shows better result in their experiment. Assuming the distribution function of noise level has the form of eq. (11):

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 \tag{11}$$

Where, x denotes the average pixel intensity of the superpixel block and a_0, a_1, $\cdots$, a_5 are the coefficients to be estimated.

Let

$$C = \sum_{i=1}^{K} [\sigma(S_i) - f(b(S_i))]^2 \tag{12}$$

Let the partial derivatives of C equal to zero, as in formula (13), the coefficients a_0, a_1, $\cdots$, a_5 can be calculated.

$$\frac{\partial(C)}{\partial a_0} = \frac{\partial(C)}{\partial a_1} = \cdots = \frac{\partial(C)}{\partial a_5} = 0 \tag{13}$$

Therefore, for the superpixel block S_i, the probability distribution function of the noise standard deviation$\sigma(S_i)$ is obtained as follows:

$$P(\sigma(S_i)) = \frac{e^{-\lambda}.\lambda^{\sigma(S_i)}}{\sigma(S_i)!} \tag{14}$$

Where $\lambda = f([b(S_i)])$, $(i = 1, 2, ..., K)$.

2.4 Splicing region detection

As one of the main techniques of unsupervised machine learning, fuzzy clustering analysis is an effective method of data analyzing and modeling. Fuzzy clustering analysis has been used in various fields effectively such as large-scale data analysis, data mining, vector quantization, image segmentation, pattern recognition, etc.

Consider that image noise is a random signal, and it has uncertainties and is subject to various factors. On the other hand, the image tampering techniques are constantly improved, and the image tampering methods are diversification, as well as the anti-forensics techniques are widely applied. These factors cause the ambiguity between the tampered image regions and the real image regions. As a result, splicing detection is very difficult. Moreover, noise in image forensic applications is generally weak and content-related. To avoid the influence of a single factor, we estimate the noise difference using two factors, noise level and noise distribution probability, and then we constructed a splicing localization algorithm by using Fuzzy c-means (FCM) clustering.

2.4.1 Establishment data sets

Let $U = \{\bar{\sigma}_{i-}, p(S_i)\}$, where $\bar{\sigma}_i$ is the noise standard deviation and $p(S_i)$ is the noise distribution probability.

2.4.2 Calculating cluster centers

To identify the splicing regions, the superpixel blocks S_i $(i = 1, 2, \cdots, K)$ of the testing image will be divided into two categories, and the objective function is as follows:

$$J(U, V) = \sum_{i=1}^{K} \sum_{q=1}^{2} \left(u_{iq}\right)^m \left(d_{iq}\right)^2 \tag{15}$$

Here $V = \{(x_1, y_1), (x_2, y_2)\}$ indicates cluster center, Urepresents the data matrix $u_{iq} \in [0, 1]$ represents the membership value of the $i-$thsample belonging to the q-th class, and $m \in [0, \infty]$ is the weighted index. According to the experimental research on clustering validity [22], the best selection interval of m is $m \in [1.5, 2.5]$. In this work, we let $m = 2.0$. Let d_{iq} denote the Euclidean distance between the $i-$th sample$t_i = \left(\sigma_i, \bar{P}(S_i)\right)$ and the $q-$th cluster center $v_q = (v_{1q}, v_{2q})$, $(q = 1, 2)$.

It can be expressed as follows:

$$d_{iq} = \sqrt{\left(\bar{\sigma}_{-i} - v_{1q}\right)^2 + \left(P(S_i) - v_{2q}\right)^2} \tag{16}$$

$J(U, V)$ indicates the weighted square sum of the distances from the samples in every category to the cluster center. The smaller $J(U, V)$ is, the smaller the in-class dispersion will be, and the better the clustering effect will be obtained. According to clustering rule, the minimum form of $J(U, V)$ is defined as follows:

$$\min\{J(U, V)\} = \min\left\{ \sum_{i=1}^{K} \sum_{q=1}^{2} \left(u_{iq}\right)^m \left(d_{iq}\right)^2 \right\} \tag{17}$$

Using the Lagrange multipliers method, we can obtain the membership value as follows:

$$u_{iq} = \left(\sum_{j=1}^{2} \left(\frac{d_{iq}}{d_{jq}} \right)^{{}^{2}/_{(m-1)}} \right)^{-1} \tag{18}$$

Where $\sum_{q=1}^{V} u_{iq} = 1$. If exists $q,\ r$ such that $d_{rq} = 0$, then $u_{rq} = 1$. Additional, for $j \neq r$, $u_{jq} = 0$. Then the cluster centers can be defined as follows:

$$\begin{cases} v_{1q} = \dfrac{\sum\limits_{i=1}^{K} \left(u_{iq}\right)^m \overline{\sigma_i}}{\sum\limits_{i=1}^{K} \left(u_{iq}\right)^m} \\ v_{2q} = \dfrac{\sum\limits_{i=1}^{K} \left(u_{iq}\right)^m P(S_i)}{\sum\limits_{i=1}^{K} \left(u_{iq}\right)^m} \end{cases}, \quad \left(v_q = \left(v_{1q}, v_{2q}\right)\right) \tag{19}$$

The cluster center V can be obtained by repeatedly iterating to optimize the objective function $J(U, V)$, until the Euclidean distance between two adjacent cluster centers is less than the threshold ε.

2.4.3 FCM clustering results

Using the FCM clustering algorithm, the dataset U is clustered into two categories. Then the dataset is indexed to the corresponding superpixel block $S_i(i = 1, 2, \ldots\ldots, K)$ to obtain the corresponding image regions. Figure 5 shows the clustering result. In Fig. 4, the red marks the splicing image regions, the green mark indicates the original regions.

Consider that the image splicing is a partial operation, we appoint the small image region as the spliced region and mark it. We define

$$B_{MN} = \begin{cases} Splicing \quad region(index1) & if \dfrac{length(index1)}{length(index1) + length(index2)} \le 0.5 \\ Splicing \quad region(index2) & otherwise \dfrac{length(index1)}{length(index1) + length(index2)} > 0.5 \end{cases} \tag{20}$$

Where $index1$ and $index2$ are the sets of the indices of superpixel blocks, respectively. $length(\bullet)$ indicates the total number of corresponding mark blocks.

In theory, the advantages of our method include two aspects. The first, we investigate the noise difference using two factors, noise level and noise distribution

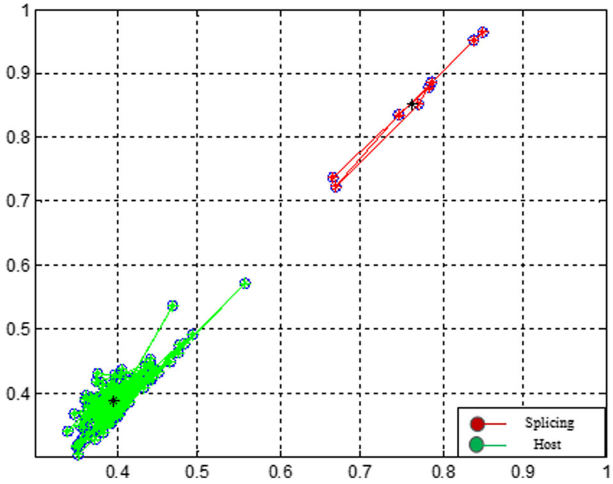


Fig. 5 Clustering results. The red marks the splicing image regions, the green mark indicates the original regions

probability. This may effectively avoid the failure of splicing localization caused by single factor. The second, for the blocks with high noise level, the probability value of the noise distribution is bigger also. Therefore, the comprehensive consideration of noise level and noise distribution probability is equivalent to amplifying the difference of noise. As a result, the splicing positioning accuracy is improved, so the proposed method has better advantages compared with other methods.

3 Experimental results and performance analysis

In this section, we will evaluate and analyze the performance of the proposed method via experiments. The experiments consist of three parts.

First, we compare the proposed method with four state-of-the-art image splicing localization methods to demonstrate the superior detection effect of the proposed method in image splicing localization, present the numerical result of pixel-level detection accuracy, and quantitatively analyze the detection effect by adding noise in spliced regions.

Second, we evaluate the robustness of the proposed method for content-preserving manipulations, such as JPEG compression, adding noise, blur, down-sampling and up-sampling, gamma correction, via quantitative indicators, and compare it with other related methods.

Third, we analyze the computational complexity of the proposed method. Computational complexity includes the calculating time spent on superpixel segmentation, noise feature extraction, and Splicing region detection. In our experiments, we test the average running time of proposed method and compare it with works [20, 24, 38] and [37].

The simulation experiments were performed on a computer with a 3.1 GHz CPU and 4 GB RAM. In experiments, we investigate the splicing detection ability, robustness, detection accuracy, and the time complexity of the proposed method, respectively; and compare with the state-of-the-art methods.

3.1 The detection effect and the detection accuracy

For the image splicing detection algorithm, one of the most important factors to evaluation the algorithm performance is the detection accuracy. In this section, we examine the detection accuracy of the proposed algorithm via visually demonstration and quantitative assessment, and compare with other related methods.

3.1.1 The visual effect of splicing detection

To evaluation the splicing detection effect of the proposed method, we test it on the Columbia uncompressed image splicing detection evaluation dataset (Columbia IPDED) [7]. In experiment, we take the original images randomly and corresponding splicing images from Columbia IPDED, then detect these images using the proposed method and methods in works [20, 24, 37], and [38], respectively. Figure 4 shows four examples of the spliced tampering images and their detection results.

In Fig. 6, the first row is the original images, and the second row is the corresponding splicing images. The third row is the detection results of the proposed method, and the spliced regions are marked with white. The fourth row is the detection results of the method [38], the spliced regions are marked with green grid, and the red grids represent the regions of error

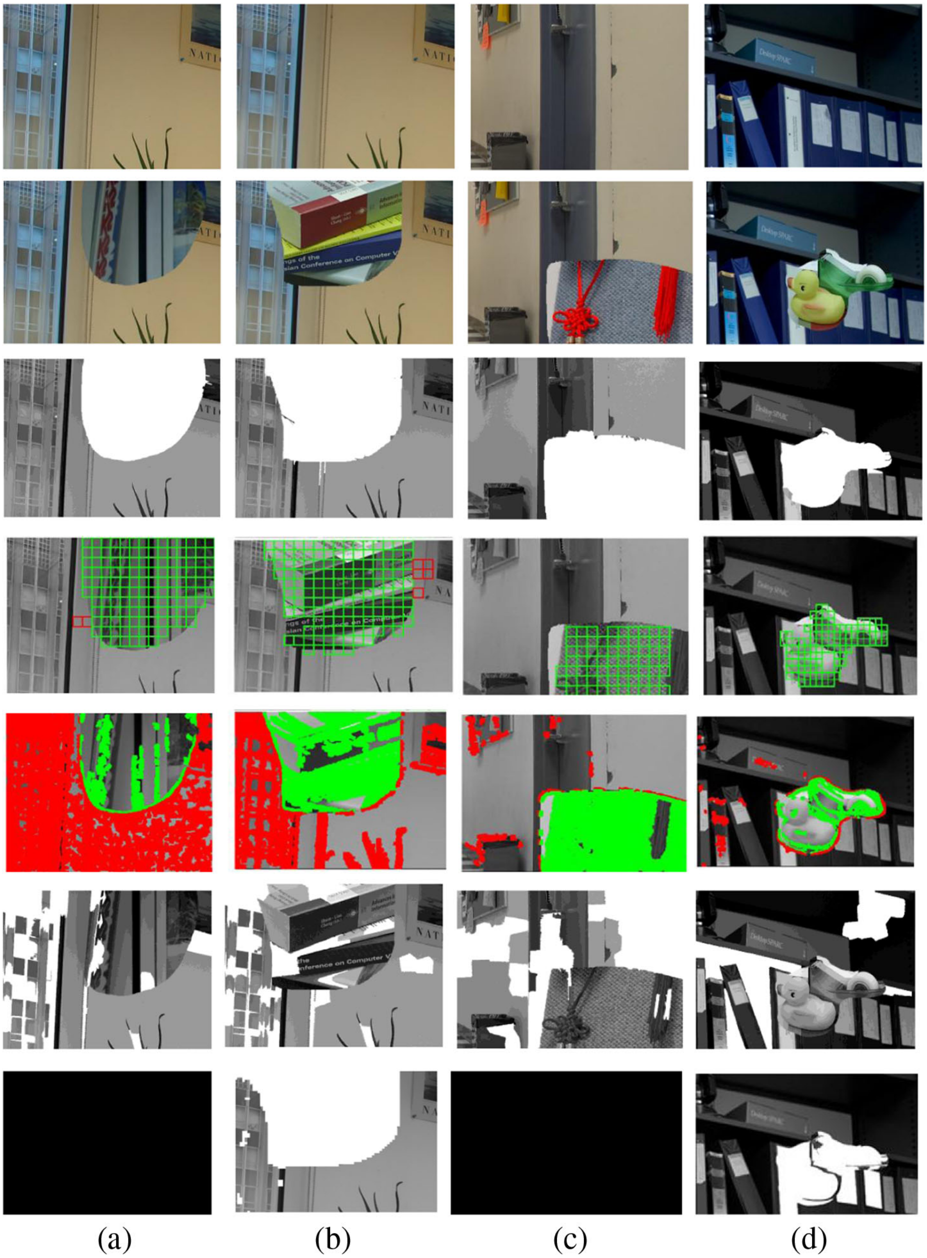


Fig. 6 Splicing detection effects and their comparison

detection. The fifth row is the detection results of the proposed method [20], and the spliced regions are marked with green, and the red grids represent the regions of error detection. The sixth row is the detection results of the proposed method [24], and the spliced regions are marked with white. The seventh row is the detection results of the proposed method [37], and the spliced regions are marked with white. As can be seen from Fig. 6, our method provides exact detection regions, whereas the other methods may fail in some cases or have more false detection. The reason is that the noise level difference between the original region and the spliced region in this dataset is typically small; the proposed method has strong distinguishing ability and can distinguish small noise differences, whereas the other methods cannot.

3.1.2 The detection accuracy

To quantitatively analysis the performance of the proposed method, we investigate the detection accuracy in pixel level. For this purpose, we formalized the true positive rate (*TPR*) and false positive rate (*FPR*) as follows.

$$TPR = \frac{N_{PS-S}}{N_{TPS}} \times 100\%$$
$$FPR = \frac{N_{PO-S}}{N_{TPO}} \times 100\%$$

Here, TPR is the rate of pixels in the spliced region that are correctly detected and FPR is the rate of image pixels in the original region that are falsely detected. $N_{PS-S}$is the number of pixels in the spliced regions that are correctly detected. N_{TPS} is the total number of pixels in splicing regions. N_{PO-S} is the number of pixels in original regions that are detected as splicing pixels. N_{TPO} is the total number of pixels in the original image. An effective splicing localization method is expected to achieve high *TPR* and low *FPR* simultaneously.

We calculate the pixel-level *TPR* and *FPR* of above testing four images, the testing results are shown in the Table 1.

As can be seen from Table 2, the proposed method provides accurate detection, and the detection accuracy is higher than those methods in [20, 24, 38] and [37].

3.1.3 Quantitative analysis of the detection accuracy

We take randomly 200 images from the BOSSbase database. First, for the original image with the size 256 × 384, add zero mean Gaussian noise with the standard deviation σ, and σ changes from 1 to 10 with step size of 1. By such a way, 10 × 200 different noise images can be obtained. We cut a 120 × 120 square region from each noise image and splice into other image to generate 10 different simulated image data sets with total 2000 images. Then we carry out the proposed algorithm to compute average TPR and FPR on each data set and compared with the works [20, 24, 38] and [37]. The results are shown in Fig. 7.

Table 2 Pixel-level accuracy and the comparative results (%)

	(a)		(b)		(c)		(d)	
	TPR	*FPR*	*TPR*	*FPR*	*TPR*	*FPR*	*TPR*	*FPR*
The proposed method	100.0	0.26	93.6	1.4	97.0	4.5	94.5	0.14
[38]	97.0	4.1	70.2	0	93.4	5.1	80.2	1.0
[20]	16.5	59.3	69.3	3.8	60.1	32.0	32.0	2.6
[24]	7.7	17.7	17.8	12.4	12.5	31.9	0	16.3
[37]	0	100	90.2	3.4	0	100	85.9	7.3

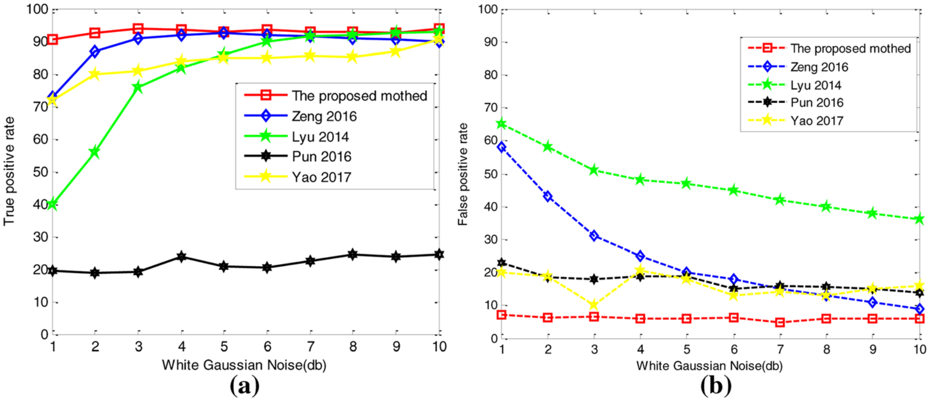


Fig. 7 The comparison of the splicing detection accuracy. **a** TPR rates comparison for image splicing localization. **b** FPR rates comparison for image splicing localization

Figure 7 shows the average TPR and FPR for all 200 images at each noise level. In the Fig. 7, the red lines represent the proposed method, the blue lines represent the method in [38], the green lines represent the method in [20], the black lines represent the method in [24], and the yellow lines represent the method in [37]. Here, the solid lines denote average TPR, and the dotted lines denote average FPR.

As can be seen from the Fig. 7, comparison with methods [20, 24, 38] and [37], not only the proposed method has higher *TPR* and lower *FPR*, also, its detection accuracy is stable and does not change with the change of noise standard deviation σ. Specially, when σ is small, the *TPR* and *FPR* of the proposed method are still satisfactory.

3.2 The analysis and comparison of the robustness

For the image splicing detection algorithm, another very important factor to evaluation the algorithm performance is the robustness. In this section, we evaluate the robustness of the proposed method via quantitative indicators, and compare it with other related methods.

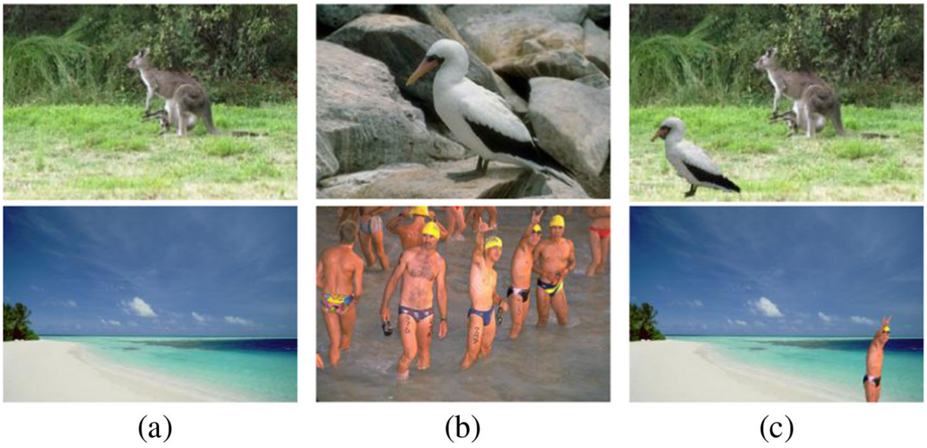


Fig. 8 The examples of the splicing images

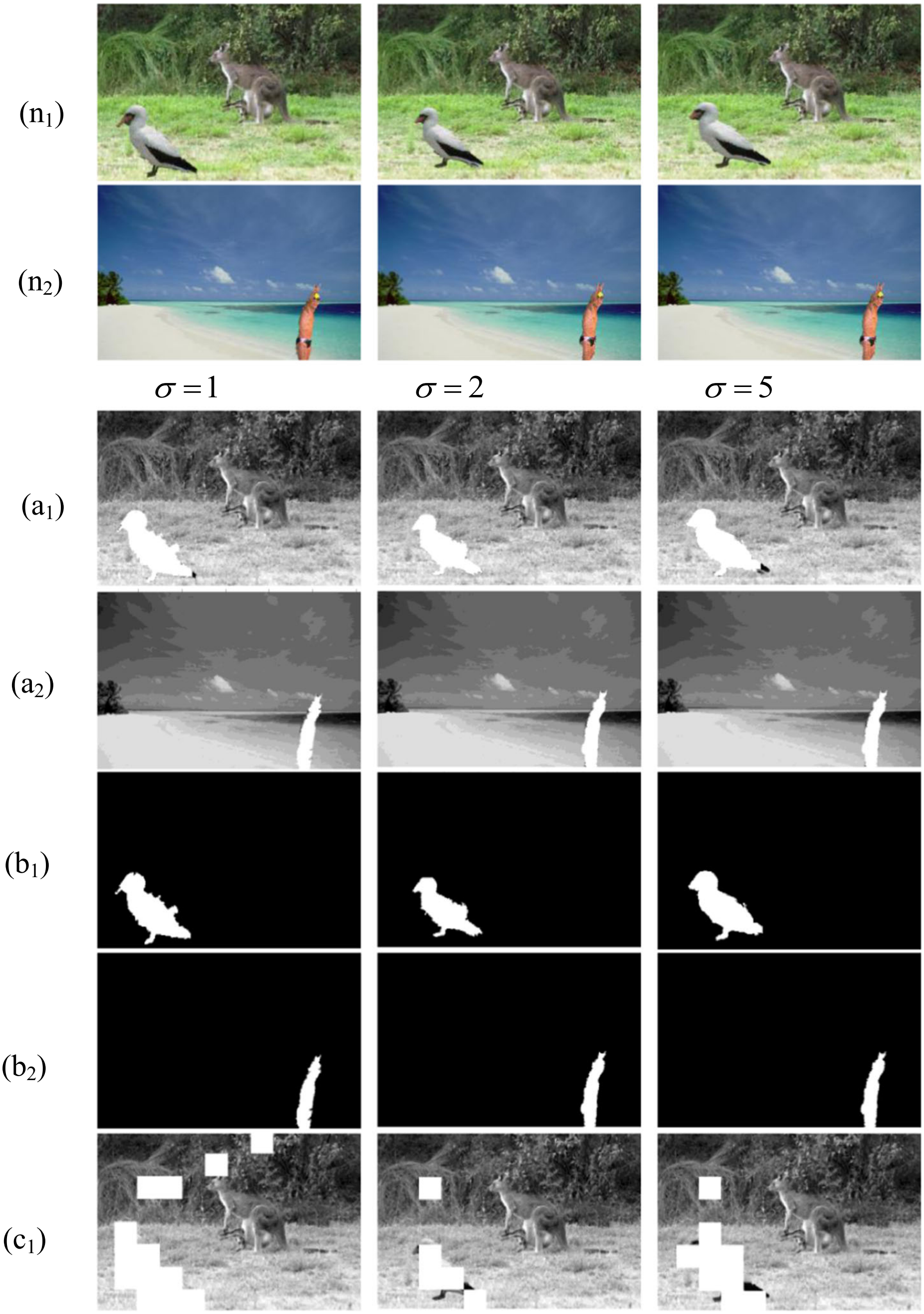


Fig. 9 The detection results and the comparison results for splicing images with different noise level

Robustness means that the algorithm is stabilized for incidental changes caused by content-preserving manipulations, such as JPEG compression, adding noise, blur, down-sampling and up-sampling, gamma correction, and so on.

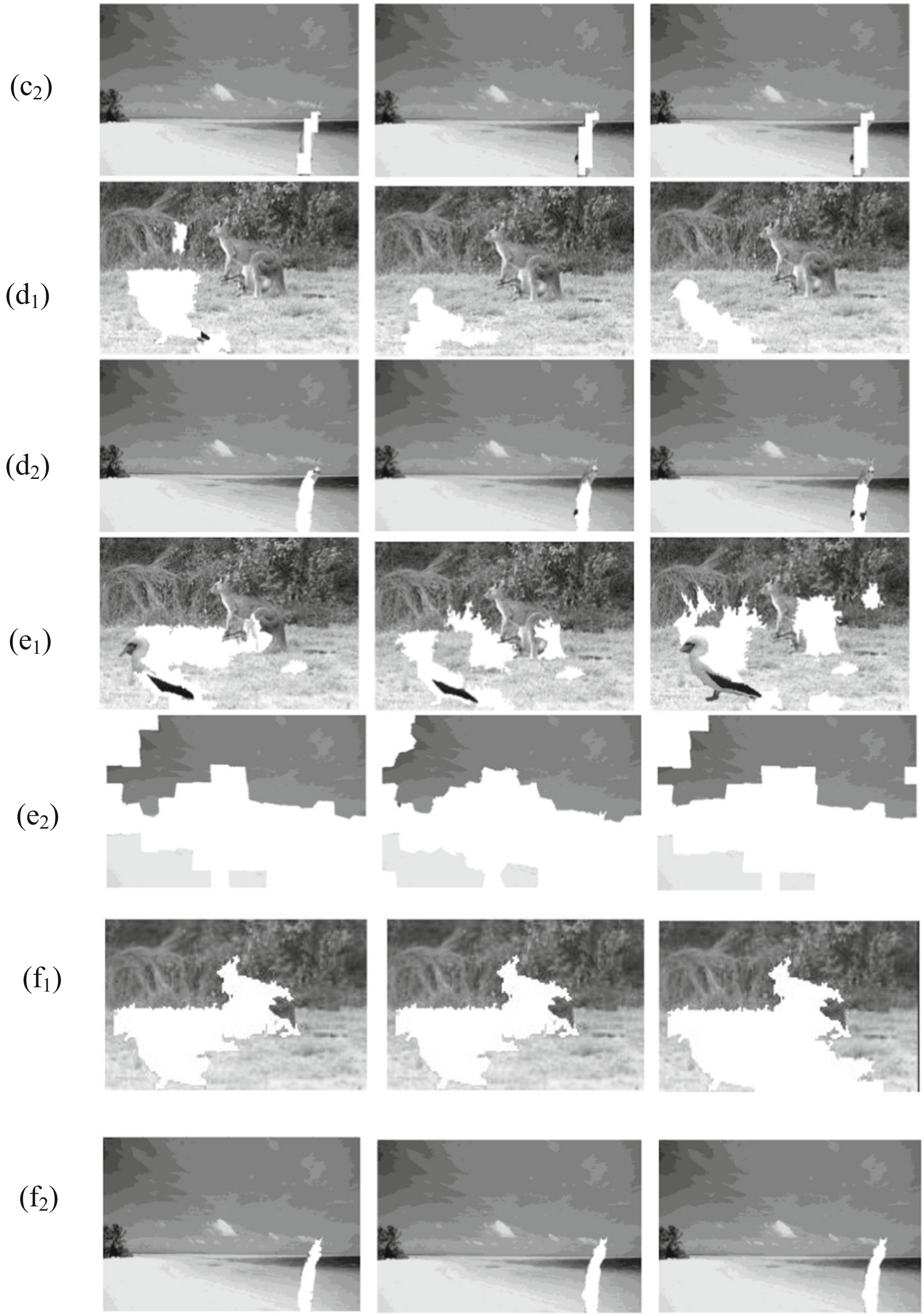


Fig. 9 (continued)

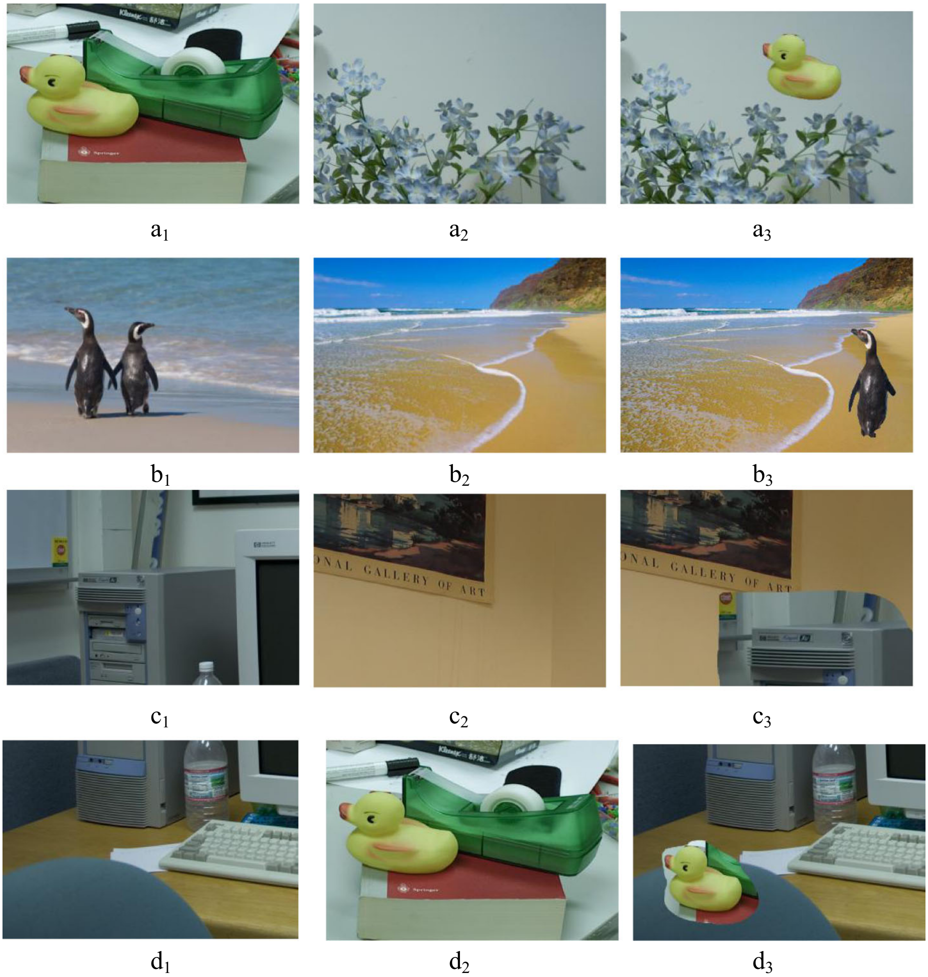


Fig. 10 The examples of the splicing images, A_1, A_2, B_1, B_2, C_1, C_2, D_1 and D_2 are original images, A_3, B_3, C_3 and D_3 are the splicing images

3.2.1 Robustness for adding noise

To investigate the robustness of the proposed method for adding noise, we firstly generate splicing image, then adding noise to the spliced image, and finally, test the splicing image using the proposed method, and comparison with the methods [20, 24, 38] and [37].

In the experiment, we randomly take 4 images from the Columbia IPDED. Figure 8 show two examples of the spliced image, in which, (a) and (b) are the original images, a bird and a man from (b) is inserted into (a) to form splicing images (c), respectively. Then the spliced images (see Fig. 8(c)) are operated by adding zero mean Gaussian noise with standard deviation σ=1, 2 and 5, respectively, as show in Fig. 9 row (n1) and row (n2). Then we perform the proposed algorithm to detection the splicing regions; and

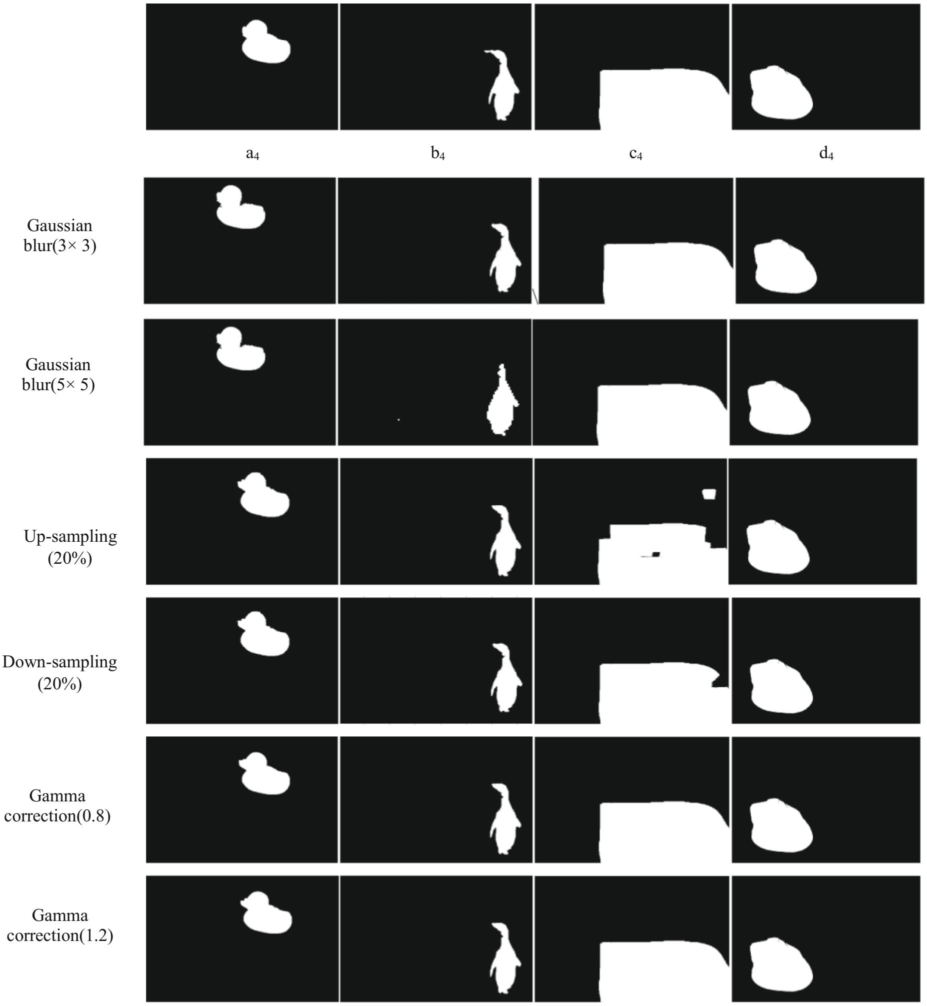


Fig. 11 The detection results of the splicing images with Post-processing, A_4, B_4, C_4 and D_4 is without post-processing operation detect results

results are shown in Fig. 9. In Fig. 9, images in row (a1) and row (a2) are the detection results of the proposed method, images in row (b1) are the binary image of the row (a1), and images in row (b2) are the binary image of the row (a2). Images in row (c1) and row (c2) are the detection results of the method [38]. Images in row (d1) and row (d2) are the detection results of the method [20]. Images in row (e1) and row (e2) are the detection results of the method [24]. Images in row (f1) and row (f2) are the detection results of the method [37]. In all the detection results, the detected splicing regions are marked with white.

As can be seen from the Fig. 9, the proposed method provides accurate detection results for splicing images, especially when the added noise is small. Whereas, method [20, 24, 38] and [37] are able to provide meaningful clues about the forgeries, but the detection results are not exact. The methods [20, 38] and [37] have higher false positives regions, and the methods [24] deteriorate more significantly and fail to locate the spliced regions.

3.2.2 Robustness for Gaussian blur, resampling and gamma correction

In order to investigate the robustness of the proposed method for Gaussian blur, resampling and gamma correction, we firstly generate splicing image, then process splicing images by post-processing operations such as, Gaussian blur, resampling and gamma correction, respectively, and finally, test those operated image using the proposed method, and comparison with the methods [20, 24, 38] and [37].

In the experiment, we randomly take 8 images from the Columbia IPDED. Figure 10 shows four examples of the spliced images, in which, A_1, A_2, B_1, B_2, C_1, C_2, D_1 and D_2 are original images, A_3, B_3, C_3 and D_3 are the splicing images. Then we perform Gaussian blur, up-sampling, down-sampling, and gamma correction for A_3, B_3, C_3 and D_3, respectively. We perform the proposed algorithm to detection the splicing regions; and results are shown under the corresponding images of the Fig. 11.

As can be seen from the Figs. 10 and 11, though the splicing images undergo post-processing, the detection results are still satisfactory.

3.2.3 The comparison of the robustness

We also compare the proposed method with the related methods through application to the whole Columbia uncompressed image splicing detection evaluation dataset before and after common post-processing. The average pixel-level quantitative comparison is reported in Table 3. As can be seen from the Table 3, comparison with the [20, 24] and [38], the proposed method is superior in robustness. However, the proposed method also shows some limitations in that it is not robust enough for JPEG compression. The reason is that the JPEG compression seriously destroys the correlation of noise, this results in a significant decrease in detection capability of the noise-based detection algorithm.

However, it can be seen from Table 3 that the [37] is less different from the TPR proposed in this paper, but the FPR the [37] is larger. Taking TPR and FPR into account, the proposed method has great advantages. The only deficiency is that literature [37] has a good advantage for JPRG compression.

Table 3 Pixel-level performance comparison (%) for the Columbia uncompressed image splicing detection valuation dataset, before and after post-processing

Method	(%)	NO post-processing	JPEG compression		Down-sampling	Gamma correction	Gaussian blur (3 × 3)
			85	95	20%	Gamma = 1	$\sigma = 1$
The Proposed Method	TPR	58.9	18.7	20.3	60.2	57.6	55.2
	FPR	7.8	4.5	3.2	9.8	10.8	11.0
Lyu et al. [20]	TPR	36.8	33.2	37.7	32.0	30.8	29.9
	FPR	23.0	24.4	20.7	20.9	27.0	28.1
Zeng et al. [38]	TPR	47.9	21.3	22.0	46.0	31.9	33.3
	FPR	18.5	11.7	10.8	23.5	17.4	19.7
Pun et al. [24]	TPR	33.9	25.4	32.1	30.3	29.8	30.2
	FPR	15.6	8.7	6.5	23.5	19.8	20.4
Yao et al. [37]	TPR	63.6	48.5	52.4	58.0	60.6	53.2
	FPR	26.5	29.8	25.0	25.6	26.9	25.8

Table 4 Computational time cost and the comparison

Method	The proposed method	[38]	[20]	[24]	[37]
Running time(s)	72.8	24.2	94.2	216.7	101

3.3 Computational complexity analysis

Computational complexity includes the calculating time spent on superpixel segmentation, noise feature extraction, and Splicing region detection. In our experiments, we test the average running time of proposed method and works [20, 24, 38] and [37] for each image from the Columbia IPDED. Table 4 shows the statistical average values. Due to the segmentation of non-overlapping blocks, the method [38] is faster. The method [24] is time-consuming because it uses multi-scale superpixel segmentation to estimate the noise level, which leads to a higher time complexity.

4 Conclusions

In this work, we propose an effective image splicing detection method. Based on the assumption that spliced regions and original regions tend to have different noise levels, we perform a block-wise local noise level estimation in the questioned image. Compared to the existing noise-based image splicing region detection methods, experimental results on different datasets have shown that the proposed method has superior performance, especially when the noise difference between the spliced region and the original region is small.

However, the proposed method shows some limitations, that is, it is not robust enough for JPEG compression, and it will invalid when the noise levels of the spliced image regions is the same with that of the source images regions.

It is worth pointing out that we cannot expect such noise-based methods to detect all potential forgeries in a real scenario, due to the variety of the image splicing. However, as stated in [20], the noise-based methods can give the forensic investigator important clues about the potential forgery when combined with other detection methods.

Acknowledgements This work was supported by the National Major Research and Development Plan Program of China under Grant No.2016YFB1001004; the National Natural Science Foundation of China under Grant No.61772416 and No. 91646108; Shaanxi province technology innovation guiding fund project, No.2018XNCG-G-02. The foundation of the State Key Laboratory of Astronautic Dynamics.

References

1. Alahmadi A, Hussain M, Aboalsamh H, Muhamma G, Bebis G (2013) Splicing image forgery detection based on DCT and local binary pattern. Proc IEEE GLOBALSIP, Austin, TX, USA: 253-256
2. Al-Hannadi M H, Hussain M, Aboalsamh H, Muhamma G, Bebis G (2013) Curvelet transform and local texture based image forgery detection. International Symposium on Visual Computing, Crete, Greece: 503–512
3. Bahrami K, Kot A C, Fan J (2013) Splicing detection in out-of-focus blurred images. Proc The IEEE International workshop on information forensics and security, Guangzhou, China: 15-21
4. Bahrami K, Alex C, Li L, Li H (2015) Blurred image splicing localization by exposing blur type inconsistency. IEEE Trans Inform Forensics Sec 5(10):999–1009

5. Cao G, Zhao Y, Ni R (2010) Edge-based blur metric for tamper detection. J Inform Hiding Multimed Signal Process 1(1):20–27
6. Chierchia G, Poggi G, Sansone C, Verdoliva L (2014) A Bayesian-MRF approach for PRNU-based image forgery detection. IEEE Trans Inf Forensics Sec 9(4):554–567
7. Columbia DVMM Research Lab (2004). Columbia Image Splicing Detection Evaluation Dataset, http://www.ee.Columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/Auth Spliced DataSet.htm
8. Cozzolino D, Verdoliva L (2017) Single-image splicing localization through autoencoder-based anomaly detection. IEEE International Workshop on Information Forensics and Security: 1–6
9. Gallagher A, Chen T (2010) Image authentication by detecting traces of demosaicing. Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops Anchorage, AK: 1–8
10. Han J G, Park T H, Yong H M, et al (2018) Quantization-based Markov feature extraction method for image splicing detection, Machine Vision & Applications, (6):1-10
11. Hsu Y, Chang S (2006) Detecting image splicing using geometry invariants and camera characteristics consistency. Proc. 2006 IEEE international conference on multimedia and expo, Toronto, Ontario, Canada: 9-12
12. Hsu Y, Chang S (2010) Camera response functions for image forensics: an automatic algorithm for splicing detection. IEEE Trans Inform Forensics Sec 5(4):816–825
13. Iakovidou C, Zampoglou M, Papadopoulos S et al (2018) Content-aware detection of JPEG grid inconsistencies for intuitive image forensics. J Visual Commun Image Represent 54:155–170
14. Johnson M K, Farid H (2005) Exposing digital forgeries by detecting inconsistencies in lighting. Proc 7th workshop on Multimedia & Security, New York, USA: 1–10
15. Lakhani G (2008) Enhancing Poisson's equation-based approach for DCT prediction. IEEE Trans Image Process Public IEEE Signal Process Soc 17(3):427–430
16. Lancaster P, Salkauskas K (1986) Curve and surface fitting. Academic Press
17. Liu Q, Sung A (2009) A new approach for JPEG resize and image splicing detection. Proc ACM Multimed Sec Workshop 23(4):716–744
18. Liu Q, Cao X, Deng C, Guo X (2011) Identifying image composites through shadow matte consistency. IEEE Trans Inf Forensics Sec 6(3):1111–1122
19. Liu X, Tanaka M, Okutomi M (2014) Practical signal-dependent noise parameter estimation from a single noisy image. IEEE Trans Image Process Publ IEEE Signal Process Soc 23(10):4361–4371
20. Lyu S, Pan X, Zhang X (2014) Exposing region splicing forgeries with blind local noise estimation. Int J Comput Vis 110(2):202–221
21. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on feature distributions. Pattern Recogn 29(1):51–59
22. Pal NR, Bezdek JC (1995) On cluster validity for the fuzzy c- means model. IEEE Tran Fuzzy Syst 3(3):370–379
23. Popescu A, Farid H (2005) Exposing digital forgeries in color filter array interpolated images. IEEE Trans Signal Process 10(53):3948–3959
24. Pun C M, Liu B, Yuan X C (2016) Multi-scale noise estimation for image splicing forgery detection. Academic press, Inc 38 (C) :195-206
25. Pyatykh S, Hesser, J (2015) MMSE estimation for Poisson noise removal in images, Computer Science
26. Pyatykh S, Hesser J, Zheng L (2013) Image noise level estimation by principal component analysis. IEEE Trans Image Process A Public IEEE Signal Process Soc 22(2):687
27. Salloum R, Ren Y, Kuo C (2017) Image Splicing Localization Using A Multi-Task Fully Convolutional Network (MFCN)
28. Salmon J, Harmany Z, Deledalle CA et al (2012) Poisson noise reduction with non-local PCA. J Math Imag Vision 48(2):279–294
29. Shah A, El-Sayed M, El-Alfy (2018) Image splicing forgery detection using DCT coefficients with multi-scale LBP. Int. Conf. Computing Sciences and Engineering (ICCSE): 1–16
30. Song C, Lin X (2014) Natural image splicing detection based on defocus blur at edges. Proc 2014 symposium on privacy and security in commutations, Shanghai, China: 24-26
31. Wang B, Kong X (2012) Image splicing localization based on re-demosaicing. In: D. Zeng (eds) Advances in Information Technology and Industry Applications. Lecture Notes in Electrical Engineering, 136, Berlin, Heidelberg: 725-732
32. Wang W, Dong J, Tan T (2009) Effective image splicing detection based on image chroma. Proc. international conference on image processing, Cairo, Egypt, Nov. 7-10: 1249-1252

33. Wang W, Dong J, Tan T (2014) Image tampering detection based on stationary distribution of Markov chain. Proc International Conference on Image Processing, Hong Kong, China: 2101-2104
34. Wang P, Liu F, Yang C et al (2018) Blind forensics of image gamma transformation and its application in splicing Detectio. J Visual Commun Image Represent 55:80–90
35. Wang P, Liu F, Yang C, et al (2018) Blind forensics of image gamma transformation and its application in splicing detection. Journal of Visual Communication & Image Representation: 81–90
36. Wattanachote K, Shih TK, Chang WL, Chang HH (2015) Tamper detection of jpeg image due to seam modifications. IEEE Trans Inf Forensics Sec 10(12):2477–2491
37. Yao H, Wang S, Zhang X, Qin C, Wang J (2017) Detecting image splicing based on noise level inconsistency. Multimed Tools Appl 6(10):1–23
38. Zeng H, Zhan Y, Kang X, et al (2016) Image splicing localization using PCA-based noise level estimation. Multimedia Tools & Applic: 1–17
39. Zhang X, Fang Z, Wang S (2009) Image splicing detection using camera characteristic inconsistency, Proc International conference on multimedia information networking and security, Washington, DC, USA: 20-24
40. Zhang W, Cao X, Zhang J, Zhu J, Wang P (2009) Detecting photographic composites using shadows. Proc IEEE international conference on multimedia and expo, New York, USA: 1042–1045
41. Zhang W, Cao X, Qu Y, Hou Y, Zhao H, Zhang C (2010) Detecting and extracting the photo composites using planar homography and graph cut. IEEE Trans Inform Forensics Sec 5(3):544–555
42. Zhang Y, Zhao C, Pi Y, Li S (2012) Revealing image splicing forgery using local binary patterns of DCT coefficients. In:proc. international conference on communications, signal processing, and systems, New York, NY, Jan.1-3. In: pp 181-189
43. Zhang Q, Lu W, Wang R et al (2018) Digital image splicing detection based on Markov features in block DWT domain. Multimed Tools Appl 7(23):31239–31260
44. Zhao X, Li J, Li S, Wang S (2010) Detecting digital image splicing in Chroma spaces. Proc. the 9th international conference on digital watermarking, Seoul Korea: 12-22

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Depeng Zhang received his B.S. degree in mathematics from Shangluo University, Shannxi, China, in 2016, and M.S. degree in mathematics from Xi'an University of Technology, Xi'an, Shannxi, China. His research interests include multimedia forensics and security, images processing.

Xiaofeng Wang received her B.S. degree in applied mathematics from Tianjin University, China, and both her M.S. degree in mathematics and her Ph.D. degree in mechanical and electronic engineering from the Xi'an University of Technology, China. In 2007, she joined the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, where she was a post-doctoral researcher until 2010. In 2012, she joined the Grasp Lab, University of Pennsylvania, where she was a visiting scholar until 2013. She is currently a professor with the Department of Mathematics, Xi'an University of Technology, China. Her current research interests include multimedia forensics and security, image processing, steganography and steganalysis.

Meng Zhang received her B.S. degree in mathematics from Xi'an University of Technology, Xi'an, Shaanxi, China, in 2016. Currently her is studying for an M.S. degree in mathematics, Xi'an University of Technology, Xi'an, Shannxi, China. Her research interests include multimedia forensics and security, Statistical learning.

Jiaojiao Hu received her B.S. degree in mathematics from Xi'an University of Technology, Xi'an, Shaanxi, China, in 2016. Currently her is studying for an M.S. degree in mathematics, Xi'an University of Technology, Xi'an, Shannxi, China. her research interests include multimedia forensics and security, Statistical learning.