Hindawi
IET Signal Processing
Volume 2024, Article ID 9929900, 11 pages
https://doi.org/10.1049/2024/9929900

Research Article

CFA-Based Splicing Forgery Localization Method via Statistical Analysis

Lei Liu,[1] Peng Sun,[2] Yubo Lang,[2] and Jingjiao Li[1]

[1]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[2]Department of Public Security Information Technology and Intelligence, Criminal Investigation Police University of China, Shenyang 110035, China

Correspondence should be addressed to Lei Liu; 380087534@qq.com

Received 19 July 2023; Revised 9 February 2024; Accepted 22 February 2024; Published 16 April 2024

Academic Editor: Richard Dansereau

The color filter array of the camera is an effective fingerprint for digital forensics. Most previous color filter array (CFA)-based forgery localization methods perform under the assumption that the interpolation algorithm is linear. However, interpolation algorithms commonly used in digital cameras are nonlinear, and their coefficients vary with content to enhance edge information. To avoid the impact of this impractical assumption, a CFA-based forgery localization method independent of linear assumption is proposed. The probability of an interpolated pixel value falling within the range of its neighboring acquired pixel values is computed. This probability serves as a means of discerning the presence and absence of CFA artifacts, as well as distinguishing between various interpolation techniques. Subsequently, curvature is employed in the analysis to select suitable features for generating the tampering probability map. Experimental results on the Columbia and Korus datasets indicate that the proposed method outperforms the state-of-the-art methods and is also more robust to various attacks, such as noise addition, Gaussian filtering, and JPEG compression with a quality factor of 90.

1. Introduction

With the rapid development of image editing technologies, digital image manipulation has become increasingly easy to perform. Unfortunately, tampered images can introduce harmful impacts through the rapid distribution on the Internet. Consequently, image forensics aimed at forgery detection, and localization or camera identification has attracted significant attention in recent years [1]. In practical forensic applications, researchers are more interested in forgery localization, i.e., locating tampered regions, rather than other goals [2].

Most forgery localization methods can be classified into physics-based methods and statistical. The physics-based methods study physical inconsistencies of images, such as the direction of incident light [3], illumination color [4], or shading and shadows [5]. These methods analyze the overall image information with physical models. They are robust to most image postprocessing, such as resizing and recompression. Although they perform well on quite controlled scenes, they are seldomly applicable to real-world images [6].

The most successful and widespread forgery localization methods are statistical. They depend on the inherent intrinsic fingerprints left on the image during the capture process, such as noise level [7, 8], lens aberration [9], or color filter array (CFA) [10, 11]. Although these efficient methods have been widely used, their localization performance degrades significantly for images undergoing postprocessing, such as median filtering.

Fortunately, most postprocessing operations can be revealed, such as resampling [12, 13], median filtering [14, 15], and contrast enhancement [16, 17]. Moreover, the various forgery localization methods are considered as tools, and a fusion framework combining different tools can avoid their drawbacks and limitations in practical applications. Fontani et al. [18] employed Dempster–Shafer theory to define a fusion framework for image forensics, which can be easily extended incrementally with new tools. Jeong et al. [19] proposed to identify the types of image forgery using a set of mixed statistical moments. Furthermore, Cozzolino et al. [20] fused the outputs of two fine-tuned algorithms to exploit their respective strengths and weaknesses. This technique obtained the best score in phase 1 of the first Image Forensics Challenge in 2013. Benefiting from the use of statistical methods as tools in fusion framework for practical applications, the improvement of single statistical method still makes sense.

In this paper, we propose a novel CFA-based forgery localization method. Most previous CFA-based methods assume that the interpolation algorithms used in digital cameras are linear, thereby simplifying the model. However, the interpolation algorithms used are often nonlinear [21], which reduces the performance of these methods in practical applications. For the nonlinear interpolation algorithms, the coefficients may vary with different image components, but the acquired pixel domain used for interpolation can be assumed constant. The interpolation process is similar to low-pass filtering making the interpolated pixel value linearly relate to the acquired pixel values in this domain. Therefore, we calculate the probability that an interpolated pixel value is within the range of its neighboring acquired pixel values within the predicted window size, which is normalized to obtain a new feature. Finally, the expectation–maximization algorithm and curvature are employed for statistical distribution analysis to obtain the tampering probability map. This method is independent of linear assumption and insensitive to content, resulting in improved performance. The experimental results show that the proposed method outperforms the reference methods and is more robust to attacks compared to other CFA-based methods.

The main contributions of this paper can be summarized as follows: (1) A content insensitive CFA fingerprint is proposed for forgery of localization. (2) Curvature is used for automatically determining whether the statistical feature can distinguish between original and tampered regions. (3) Experiments using publicly available datasets show that the proposed method outperforms the reference methods.

This work has been organized as follows. Section 2 reviews the previous works of CFA in the image forensics task. In Section 3, we present the theory of the novel CFA-based forgery localization method. We describe the experiment evaluation in Section 4 and conclude this work in Section 5.

2. Related Works

Commercial digital cameras are equipped with a CFA in front of the image sensor to capture images with only one single color sample at each pixel location. In order to obtain a three-channel color image, an interpolation algorithm is employed to estimate the other two color samples. For the most widely used Bayer CFA, the green pixels are sampled on a quincunx lattice, the red and blue pixels are sampled on the complementary locations. This CFA has four configurations: RGGB, BGGR, GRBG, and GBRG. The top-left of the CFA image with the RGBG configuration is illustrated in Figure 1.

Let us suppose that $S(x,y)$, with $(x,y) \in \mathbb{Z}^2$, is the observed CFA image, and $S_G(x,y)$ denotes the acquired green signal constructed from $S(x,y)$ as follows:

$$S_G(x,y) = \begin{cases} S(x,y) & x+y, \text{ even} \\ 0 & x+y, \text{ odd} \end{cases}. \tag{1}$$

$g_{1,1}$	$r_{1,2}$	$g_{1,3}$	$r_{1,4}$	$g_{1,5}$	$r_{1,6}$	
$b_{2,1}$	$g_{2,2}$	$b_{2,3}$	$g_{2,4}$	$b_{2,5}$	$g_{2,6}$	
$g_{3,1}$	$r_{3,2}$	$g_{3,3}$	$r_{3,4}$	$g_{3,5}$	$r_{3,6}$	
$b_{4,1}$	$g_{4,2}$	$b_{4,3}$	$g_{4,4}$	$b_{4,5}$	$g_{4,6}$	...
$g_{5,1}$	$r_{5,2}$	$g_{5,3}$	$r_{5,4}$	$g_{5,5}$	$r_{5,6}$	
$b_{6,1}$	$g_{6,2}$	$b_{6,3}$	$g_{6,4}$	$b_{6,5}$	$g_{6,6}$	
			...			

Figure 1: Top-left portion of a CFA image obtained from the GRBG Bayer configuration.

The green channel $G(x,y)$ of a complete color image is composed by acquired component and interpolated component:

$$G(x,y) = \begin{cases} S(x,y) & x+y, \text{ even} \\ \sum\limits_{\mu,\upsilon=-N}^{N} \alpha_{\mu,\upsilon} S_G(x+\mu, y+\upsilon) & x+y, \text{ odd} \end{cases}, \tag{2}$$

where $\alpha_{\mu,\upsilon}$ denotes interpolation coefficients for the acquired pixels within the $(2N+1) \times (2N+1)$ window.

The specific correlations introduced by CFA interpolation can be quantified for image forensics. Popescu and Farid [10] introduced the expectation–maximization (EM) algorithm to estimate the interpolation coefficients and obtained the probability of each pixel being correlated with its adjacent pixels. The periodicity of the possibility map deriving from the interpolation artifacts presented are particularly prominently in the Fourier domain. Bammey et al. [22] found a least square optimal filter instead of the iterative EM algorithm. Furthermore, Fernández et al. [23] estimated the interpolation coefficients with the ordinary least squares algorithm and applied the discrete cosine transform on small blocks for forgery localization. The main advantage of these methods is that a wide range of modifications can be detected without previous training and knowledge. However, they rely on the estimation of interpolation coefficients, which significantly increases the computational burden.

In addition, Choi et al. [24] defined different neighbor patterns and estimated the CFA pattern with the number of intermediate values in each channel. Moreover, they measured the hue changing by the intermediate value counting approach to identify the image color modification [25]. Shin et al. [26] identified the CFA pattern configuration based on the relationship of the variance of acquired and interpolated samples in the red, blue, and green channels. Jeon et al. [21] differentiated the CFA pattern by the truncated sum of the singular values. Besides, the prediction error is most widely used, which is defined as follows [27]:

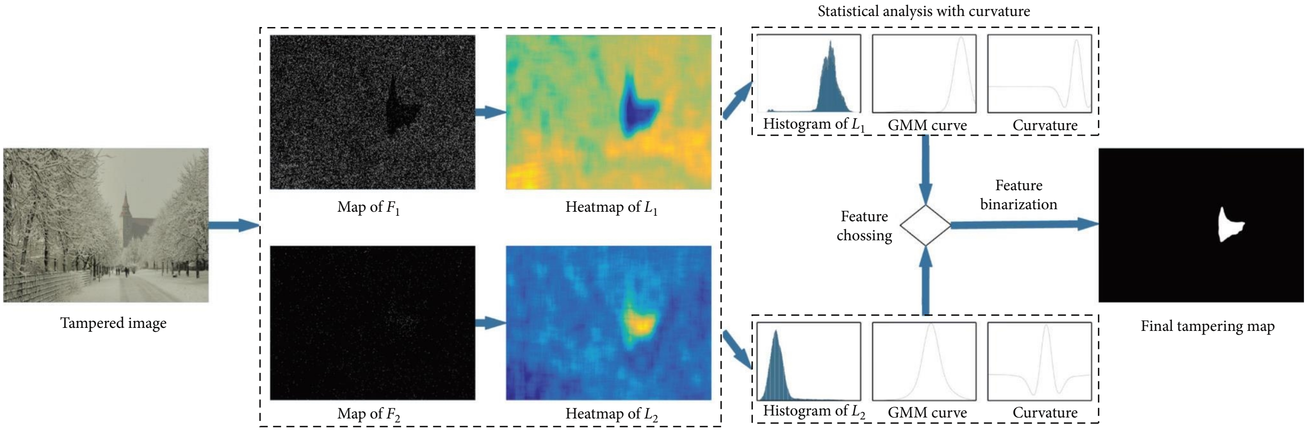


FIGURE 2: The framework of proposed CFA-based forgery localization method.

$$e(x,y)=\begin{cases} S_G(x,y)-\sum\limits_{\mu,\upsilon=-N_p}^{N_p}\alpha'_{\mu,\upsilon}S_G(x+\mu,y+\upsilon) & x+y,\ \text{even} \\ \sum\limits_{\mu,\upsilon=-N}^{N}\alpha_{\mu,\upsilon}S_G(x+\mu,y+\upsilon)-\sum\limits_{\mu,\upsilon=-N_p}^{N_p}\alpha'_{\mu,\upsilon}S_G(x+\mu,y+\upsilon) & x+y,\ \text{odd}\end{cases}, \tag{3}$$

where $\alpha'_{\mu,\upsilon}$ denotes the predicted interpolation coefficients for the acquired pixels within a $(2N'+1)\times(2N'+1)$ window.

Ferrara et al. [27] proposed a feature based on the prediction error variance to measure the absence and presence of CFA traces to obtain a fine-grained tampering possibility map that can detect small forgery. Singh et al. [28] introduced Markov random process to reduce the false detections and computational complexity on the basic study of Ferrara et al. [27]. Lu et al. [29] applied broad first search neighbors clustering algorithm to detect copied regions and duplicated regions in the copy–move images. Then they localized duplicated regions based on the prediction error. Furthermore, Chang et al. [30] detected photographic images and identified device classes based on the Fourier spectrum of the prediction error variances.

Although these methods based on prediction error have achieved good performance in various image forensics tasks, their linear interpolation assumption degrades their performance in practical applications. Most of the interpolation algorithms used in cameras are nonlinear, and their coefficients vary with the gradient to enhance edge information. As a result, these previous methods are sensitive to the content and sometimes even fail to extract CFA fingerprints effectively.

3. The Proposed Method

Similar to most previous CFA-based splicing forgery localization methods, we study the familiar Bayer CFA in the green channel. For each square of the green channel, the number of acquired and interpolated pixels is equal. These two kinds of pixels can be decomposed according to even and odd locations. However, the interpolated pixels have four locations in red and blue channels. Consequently, CFA feature extraction by applying the green channel can effectively reduce computation complexity. The proposed forgery localization framework is illustrated in Figure 2.

Let $G(x_0,y_0)$ be the pixel value at (x_0,y_0) of $G(x,y)$. Equation (2) shows that the interpolated pixel value is a weighted sum of its neighboring acquired pixel values, and the weights have:

$$\sum_{\mu,\upsilon=-N}^{N}\alpha_{\mu,\upsilon}=1\ . \tag{4}$$

Let N_r be the real N used in the interpolation algorithm of the camera. For example, N_r is equal to 1 for the bilinear interpolation algorithm and N_r is equal to 2 for gradient-based interpolation algorithm [10].

Let $Q_N(x_0,y_0)$ denote the values of the pixels at the quincunx lattice centered of $G(x_0,y_0)$ within the $(2N+1)\times(2N+1)$ window. The minimum and maximum values of $Q_N(x_0,y_0)$ is defined as follows:

$$\begin{aligned}\mathrm{Min}^N_{x_0,y_0}&=\mathrm{Min}(Q_N(x_0,y_0))\\ \mathrm{Max}^N_{x_0,y_0}&=\mathrm{Max}(Q_N(x_0,y_0))\end{aligned}\ . \tag{5}$$

When $G(x_0,y_0)$ is the interpolated pixel and $N=N_r$, we can conclude that $G(x_0,y_0)$ ranges from $\mathrm{Min}^N_{x_0,y_0}$ to $\mathrm{Max}^N_{x_0,y_0}$:

$$\mathrm{Min}^N_{x_0,y_0}\le G(x_0,y_0)\le \mathrm{Max}^N_{x_0,y_0}\ . \tag{6}$$

The probability that $G(x_0,y_0)$ satisfies Equation (4) is defined as P_{int}. When $G(x_0,y_0)$ is the acquired pixel, P_{int} is denoted as P_A; when $G(x_0,y_0)$ is the interpolated pixel, P_{int} is denoted as P_I. Obviously, $P_A<1$ and $P_I=1$.

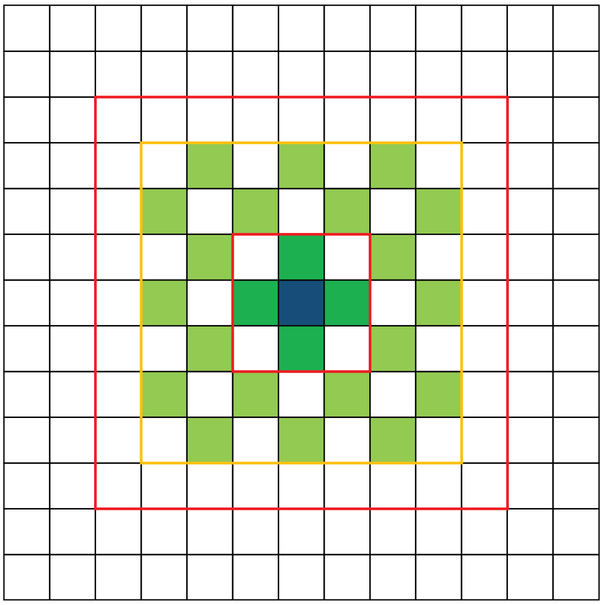


Figure 3: The blue cell indicates $G(x_0, y_0)$ and the green cells indicate $Q_N(x_0, y_0)$, where $N = 3$. The dark green cells have larger interpolation coefficients. The yellow window indicates the real window of interpolation, $N_r = 3$; the red window indicates the two prediction windows, the small one with $N_p = 1$ and the large one with $N_p = 4$.

Generally, since the in-camera interpolation algorithm is unknown, N_r is also unknown. Therefore, the predicted window size is used, which is named N_p. P_I can have various states with different relationships between N_r and N_p.

As shown in Figure 3, the yellow window denotes the real window including the acquired pixels used for interpolation, namely $N_r = 3$. The red windows denote the predicted windows for interpolation, i.e., $N_p = 1$ and $N_p = 4$. Moreover, the dark green cells denote bigger coefficients, and the pale green cells denote smaller coefficients for the interpolation. For the bigger red window ($N_p \geq N_r$), $Q_N(x_0, y_0)$ contains all acquired pixel values used for interpolation, and $G(x_0, y_0)$ is linearly correlated to it, resulting in $P_I = 1$. However, for the smaller red window ($N_p < N_r$), some of the acquired pixel values used for interpolation are not within $Q_N(x_0, y_0)$, resulting in $P_I < 1$.

For most interpolation algorithms, the acquired pixel values closest to the interpolated pixel have higher weights. These neighboring values contribute significantly to the interpolated value. Therefore, when $N_p < N_r$, $G(x_0, y_0)$ and $Q_N(x_0, y_0)$ are still strongly correlated and $P_A < P_I$, which can be used to distinguish between interpolated pixels and acquired pixels. Additionally, since P_I is mainly affected by the difference between N_r and N_p, it is constant for the same interpolation algorithm. Specifically, P_I can be used to differentiate various interpolation algorithms, and it is insensitive to the content.

To obtain P_{int}, we define the comparison result $Cp(x_0, y_0)$ as follows:

$$Cp(x_0, y_0) = \left(G(x_0, y_0) - \text{Min}^N_{x_0, y_0}\right)\left(\text{Max}^N_{x_0, y_0} - G(x_0, y_0)\right) , \tag{7}$$

where $N = N_p$. When the $G(x_0, y_0)$ satisfies Equation (6), $Cp(x_0, y_0) \geq 0$. When the $G(x_0, y_0)$ does not satisfy Equation (6), $Cp(x_0, y_0) < 0$.

Since the locations of the acquired and interpolated pixels are unknown, P_{int} needs to be estimated on the even and odd locations. For the green channel of an $M \times M$ image:

$$F_1(x_0, y_0) = \begin{cases} 1, & x + y \text{ odd and } Cp(x_0, y_0) \geq 0 \\ 0, & \text{others} \end{cases} , \tag{8}$$

$$F_2(x_0, y_0) = \begin{cases} 1, & x + y \text{ even and } Cp(x_0, y_0) \geq 0 \\ 0, & \text{others} \end{cases} . \tag{9}$$

F_1 and F_2 are two obtained binarized comparison result maps whose densities can be used to estimate P_A and P_I, respectively. Binarized comparison result maps F_1 and F_2 are divided into $b \times b$ sub-blocks at one-pixel step, and the sums of these values in each block are denoted as $B_1(x_0, y_0)$ and $B_2(x_0, y_0)$, respectively. The density of F_1 and F_2, named L_1 and L_2, are estimated by the following equations:

$$L_1(x_0, y_0) = \frac{B_1(x_0, y_0)}{b^2} , \tag{10}$$

$$L_2(x_0, y_0) = \frac{B_2(x_0, y_0)}{b^2} . \tag{11}$$

To establish a simple and tractable model, we assume that L_1 and L_2 are Gaussian distribution in the original image. For the L_1 of a forgery image, let M_1 and M_2 be the hypotheses of the original and tampered regions. Since the CFA fingerprints in M_1 and M_2 are different, we can describe pixels belonging to M_1 and M_2 with the conditional probability density functions as follows:

$$\Pr(L_1 | M_1) = \mathcal{N}(\mu_1, \sigma_1^2) , \tag{12}$$

$$\Pr(L_1 | M_2) = \mathcal{N}(\mu_2, \sigma_2^2) , \tag{13}$$

where μ_1 and μ_2 are different, making the distribution of $L_1(x_0, y_0)$ have two peaks, which can be regarded as a Gaussian mixture model (GMM).

To analyze the distribution of L_1, we introduce the EM algorithm [31]. It is a famous iterative method to estimate the means (μ_1 and μ_2), variances (σ_1 and σ_2) and mixing coefficients (π_1 and π_2) of the component distributions by maximizing the expectation of a complete log-likelihood function. With these parameters, the GMM can be written as follows [32]:

$$Y(t|\lambda) = \pi_1 \mathcal{N}(t|\mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(t|\mu_2, \sigma_2^2) , \tag{14}$$

where $Y(t|\lambda)$ is a GMM function fitted by parameters, $\lambda = \{\pi_1, \mu_1, \sigma_1, \pi_2, \mu_2, \sigma_2\}$, and for notational simplicity, we denote it by y_t, t is a 1D continuous-valued data vector, $\mathcal{N}(t|\mu_1, \sigma_1^2)$ and $\mathcal{N}(t|\mu_2, \sigma_2^2)$ are the component Gaussian densities. However, for the original image, Y_t is assumed to be a Gaussian distribution with only one peak. Therefore, we introduce the curvature of Y_t to distinguish between GMM and Gaussian distributions:

$$K_t = \frac{\left|Y_t^{(1)}\right|}{\left(1 + Y_t^{(2)}\right)^{3/2}}, \tag{15}$$

where $Y_t^{(1)}$ and $Y_t^{(2)}$ are the first-order and second-order derivatives of Y_t. For the Gaussian distribution, the curvature changes from negative to positive and then to negative. Therefore, the curvature of the Gaussian distribution has two positive and negative changes, while the curvature of the GMM has more than three changes. The times of positive and negative changes in K_t are counted and marked with Lab_1:

$$Lab_1 = \begin{cases} 1, & \mathrm{Num}_k \geq 3 \\ 0, & \mathrm{Num}_k < 3 \end{cases}, \tag{16}$$

where Num_k is the times of positive and negative changes in K_t. When $Lab_1 = 1$, the distribution of L_1 has two peaks, assuming a GMM distribution. Otherwise, the distribution of L_1 has only one peak, assuming a Gaussian distribution.

In the same way, we can get Lab_2 from L_2. Ultimately, we choose the appropriate feature as the tampering probability map through Lab_1 and Lab_2. When $Lab_1 = 1$ and $Lab_2 = 0$, L_1 is used; when $Lab_1 = 0$ and $Lab_2 = 1$, L_2 is used; when $Lab_1 = 1$ and $Lab_2 = 1$, both L_1 and L_2 can be used, and we choose to use L_1 empirically.

4. Experiment Evaluation

In this section, we conduct some experiments to evaluate the performance of the proposed method. The experimental evaluation contains Columbia Uncompressed Image Splicing Detection Evaluation Dataset (Columbia dataset [33]) and Realistic Tampering Dataset (Korus dataset [34]). The Columbia dataset was acquired using four cameras (Canon G3, Nikon D70, Canon 350D Rebel XT, and Kodak DCS 330), 15% of which were taken outdoors. The captured images from two cameras were spliced to obtain 30 tampered images, for a total of six combinations to get 180 spliced tampered images. The sizes of these forgery images range from 757×568 to $1,152 \times 768$ and the number of pixels in the tampered region is relatively large. The Korus dataset contains 220 realistic forgeries created by hand in modern photo-editing software (GIMP and Affinity Photo) and covers various challenging tampering scenarios involving both object insertion and removal. The original images were captured by four different cameras (Sony alpha57, Canon 60D, Nikon D7000, and Nikon D90) and the final forgery images are $1,920 \times 1,080$ px. Both datasets suffer a single image manipulation without any postprocessing and are saved in TIFF uncompressed format, which is beneficial to preserve the image CFA features. We only considered the reference methods that do not require training or other prior information, including CFA1 [27], CFA2, CFA3 [35], BLK [36], CAGI [37], NOI1 [38], and NOI5 [39]. For more details of the reference methods and source codes, please refer to Zampoglou et al.'s [40] study.

4.1. Performance Criteria. Forgery localization can be regarded as a special segmentation task, dividing each pixel into original (background) or tampered (foreground). Among the various evaluation criteria for segmentation tasks, mean intersection over union (MIoU) is the standard and most frequently used one [41]. It is the ratio between the intersection and the union of two sets, defined as follows:

$$\mathrm{MIoU} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN} + \mathrm{FP}}\right), \tag{17}$$

where TP, TN, FN, and FP are statistics of the observed true positives, true negatives, false negatives, and false positives, respectively.

Another important criterion is the mean pixel accuracy (MPA), the ratio of correct pixels is computed on a per-class basis and then averaged over the total number of classes:

$$\mathrm{MPA} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\right). \tag{18}$$

At last, we evaluate the performance with the Matthews correlation coefficient (MCC), the cross-correlation coefficient between the decision map and the ground truth, defined as follows:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN}\text{-}\mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}. \tag{19}$$

The MCC is robust to unbalanced classes. For some forgery images on the Korus dataset, the tampered region is much smaller than the original one, making it more appropriate to evaluate the performance of various methods with MCC.

Since the criteria used work on binary maps, and most methods only produce heatmaps with continuous values, a threshold is needed to convert these heatmaps to the corresponding binary maps. However, a single threshold algorithm will bias the detection results of different methods. Therefore, the threshold maximizing the criteria is taken. In addition, some methods just distinguish between original and tampered regions, and thus the output heatmap may have an inverted polarity with the ground truth. Consequently, we consider both the original and inverted truth ground images, leaving the best image as the result.

Most previous work has averaged the criterion scores over all test images to evaluate method performance on the dataset, such as the averageMIoU score. However, it just gives a general survey of the results on the dataset. For the sake of discussion completeness, we propose the efficiency ratio E based on the MIoU scores on the dataset:

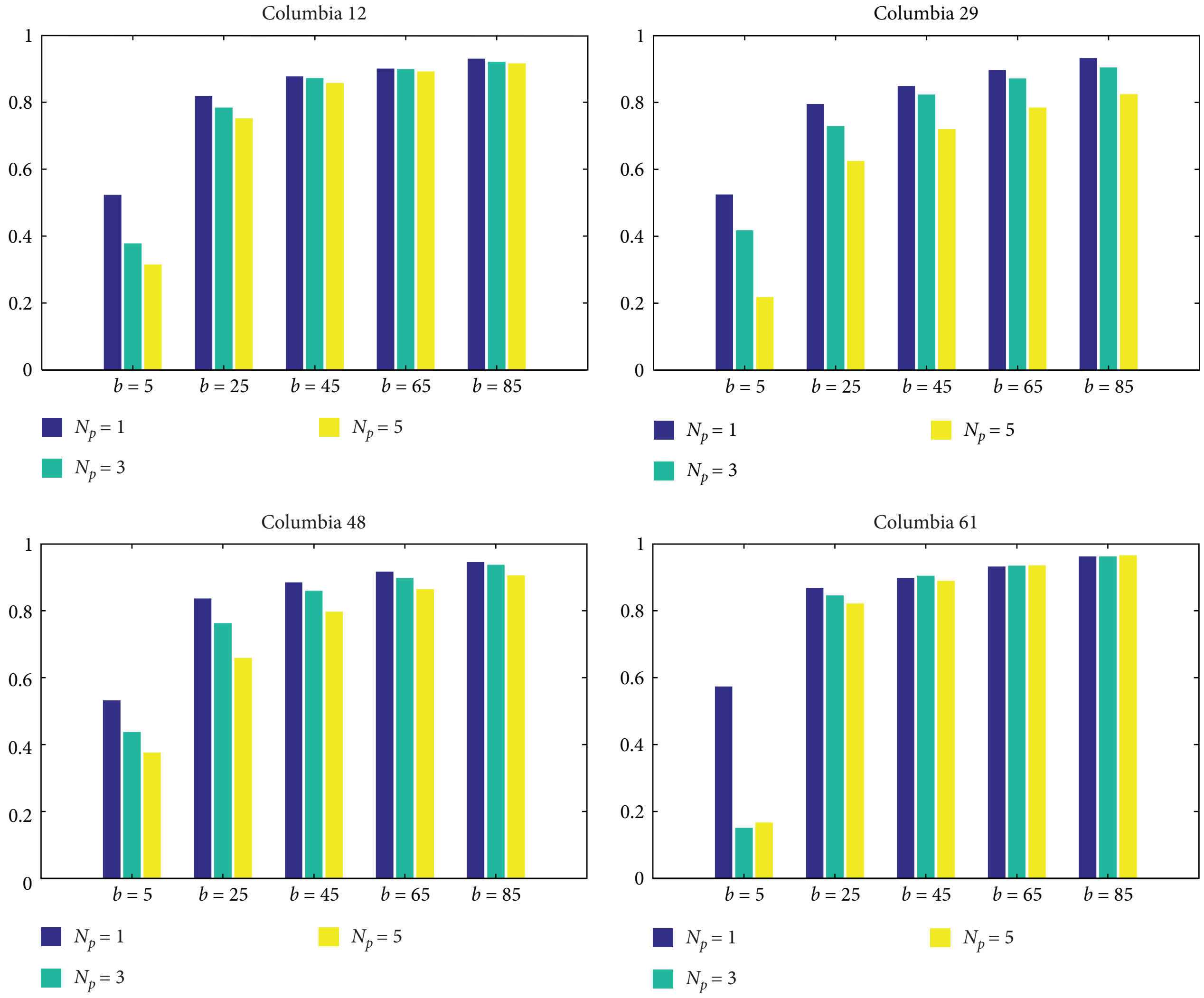


Figure 4: Comparison of peak MIoU scores for four tampered images from the Columbia dataset. Different block sizes b and prediction window sizes N_p are applied to the proposed approach. Columbia 12, 29, 48, and 61 represent the 12th, 29th, 48th, and 61st tampered images in the Columbia dataset, respectively.

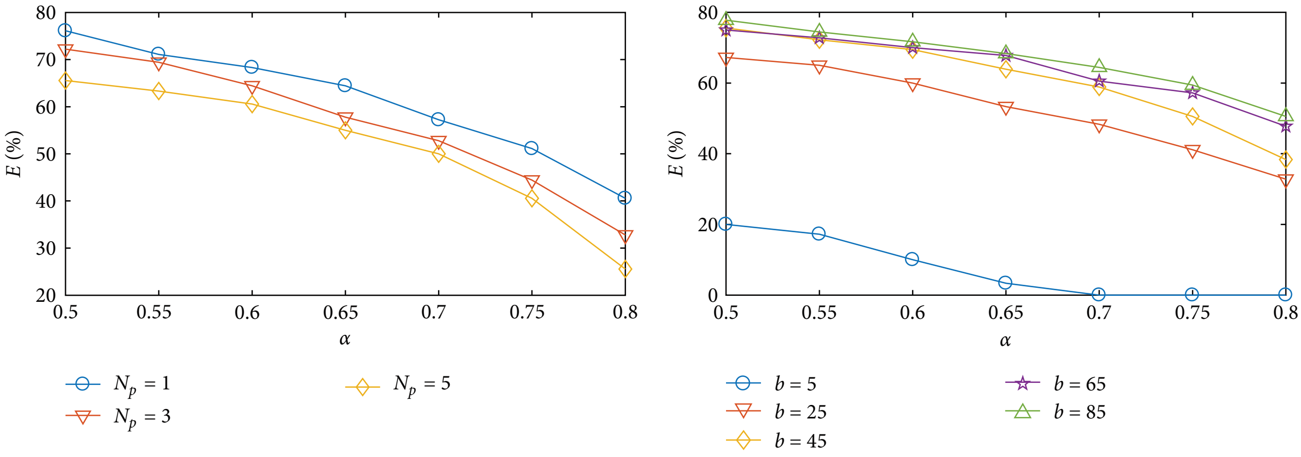
$$E = \frac{\mathrm{Count}(\mathrm{MIoU} > \alpha)}{\mathrm{Count}(\mathrm{all})} \times 100\% \,, \tag{20}$$

where $\mathrm{Count}(\mathrm{all})$ is the total number of the test images in the experiment. $\mathrm{Count}(\mathrm{MIoU} > \alpha)$ is the number of results greater than the valid threshold α. Therefore, we can set the results of $\mathrm{Count}(\mathrm{MIoU} > \alpha)$ to be valid and evaluate the detection results more precisely by controlling α.

4.2. Parameter Discussion. The proposed method is impacted by two parameters N_p and b. In this case, we assess the effect of three prediction window sizes $N_p = 1, 3, 5$. Additionally, to assess the impact of b for the proposed method, we evaluate the performance for five block sizes: 5, 25, 45, 65, and 85. To speed up the computation, we apply the Columbia dataset, which has a lower image resolution compared to the Korus dataset, and measure the performance with MIoU scores and E.

Figure 4 represents the MIoU scores of four forgery images when the method employs different parameters. For these four detection results, the MIoU scores of the detected results become higher when the block size increases. The best results are obtained in this experiment when $N_p = 1$ and $b = 85$. It is worth noting that the improvement of method performance when $b = 85$ over $b = 65$ is small. However, when $b = 85$, it increases the computational effort of the method, therefore we set b to 65 instead of 85 in our subsequent experiments.

To evaluate the impact of parameters in detail, we first evaluate the performance of the proposed method on the Columbia dataset when b is 65 and N_p takes different values as in the previous experiment. Figure 5(a) shows the efficiency ratio E at different threshold α for the three predicted



(a)

(b)

Figure 5: (a, b) Comparison of efficiency ratio E with different prediction window sizes N_p and block sizes b.

Table 1: Experiment results for Columbia and Korus datasets.

Dataset		Columbia			Korus	
Criteria	MIoU	MPA	MCC	MIoU	MPA	MCC
Ours	0.7659	0.8561	0.7255	0.6193	0.7814	0.4247
CFA1	0.6914	0.7806	0.6052	0.6154	0.7403	0.3957
CFA2	0.5776	0.7151	0.4554	0.5368	0.6726	0.2328
CFA3	0.7542	0.8507	0.7193	0.6155	0.7588	0.4055
BLK	0.4869	0.6384	0.2957	0.4935	0.6278	0.1352
CAGI	0.5817	0.7396	0.4706	0.5169	0.706	0.2229
NOI1	0.5011	0.6558	0.3263	0.5136	0.6504	0.194
NOI5	0.4041	0.6031	0.2119	0.3695	0.5793	0.0734

window sizes. At each α, the proposed method with the predicted window size of 1 outperforms the other two sizes. For example, at the valid threshold of 0.5, the E achieves 76.11% when the predicted window size is 1, whereas the E achieves 72.22% when the predicted window size is 3. At the valid threshold of 0.8, the E achieves 40.55% when the predicted window size is 1, whereas the E achieves 32.77% when the predicted window size is 3. The result of this experiment shows that the proposed method performs better with small predicted window size. Therefore, the N_p used in the proposed method should be set to 1.

We follow the same protocol for the proposed method to assess the impact of block size b. In this case, we evaluate the performance of the proposed method on the Columbia dataset when N_p is 1 and b takes different values as in the previous experiment. Figure 5(b) shows the efficiency ratio E at different valid threshold for the five block sizes. We can observe that the method performs poorly when the block size is 5 and performs particularly well when the block size is 65 and 85. In addition, when b is larger than 65, the increase of b only slightly improves the method performance. Finally, the b recommendation used in the proposed method is set to 65.

4.3. Comparative Experiments. We compare the performance of the proposed method with the reference methods with three criteria on two datasets. To evaluate the comprehensive performance of all methods, we conduct extensive experiments on one dataset by using three criteria at a time. For the proposed method, N_p is set to 1 and b is set to 65. Table 1 shows the results with respect to averageMIoU, averageMPA, and averageMCC on Columbia and Korus dataset.

We start our evaluation with a comparison of the overall performance on the two datasets. Notably, for the proposed method, the averageMIoU score on the Columbia dataset is 22.67% better than that on the Korus dataset; the average MCC score on the Columbia dataset is 70.82% better than that on the Korus dataset. In fact, the complexity of the scenario on the Korus dataset makes the test challenging. All methods achieve much worse performance on this dataset than on the Columbia dataset. Additionally, small tampered regions on the Korus dataset affect the effectiveness of MIoU and MPA. Therefore, it is reasonable to evaluate the performance of the Korus dataset with MCC, which is robust to unbalanced classes. The Columbia dataset, with large tampered regions, can be assessed with the widely used MIoU. Regardless of the criteria, the proposed method ranks first on both datasets.

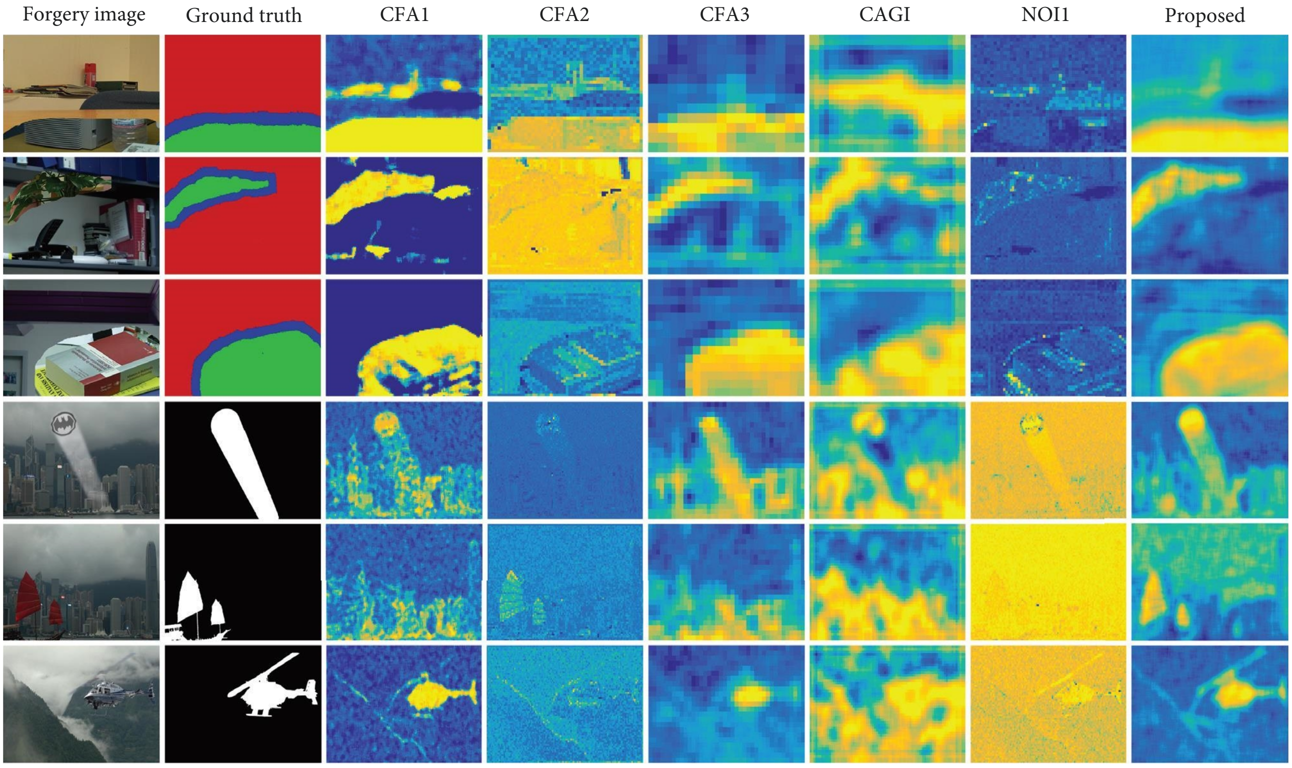


Figure 6: Example detection heatmaps from the Columbia dataset and Korus dataset. The 1st–3rd rows show the detection heatmaps from the Columbia dataset. The 4th–6th rows show the detection heatmaps from the Korus dataset. From left to right: forgery image, ground truth, and heatmaps from the six methods: CFA1, CFA2, CFA3, CAGI, NOI1, and proposed method.

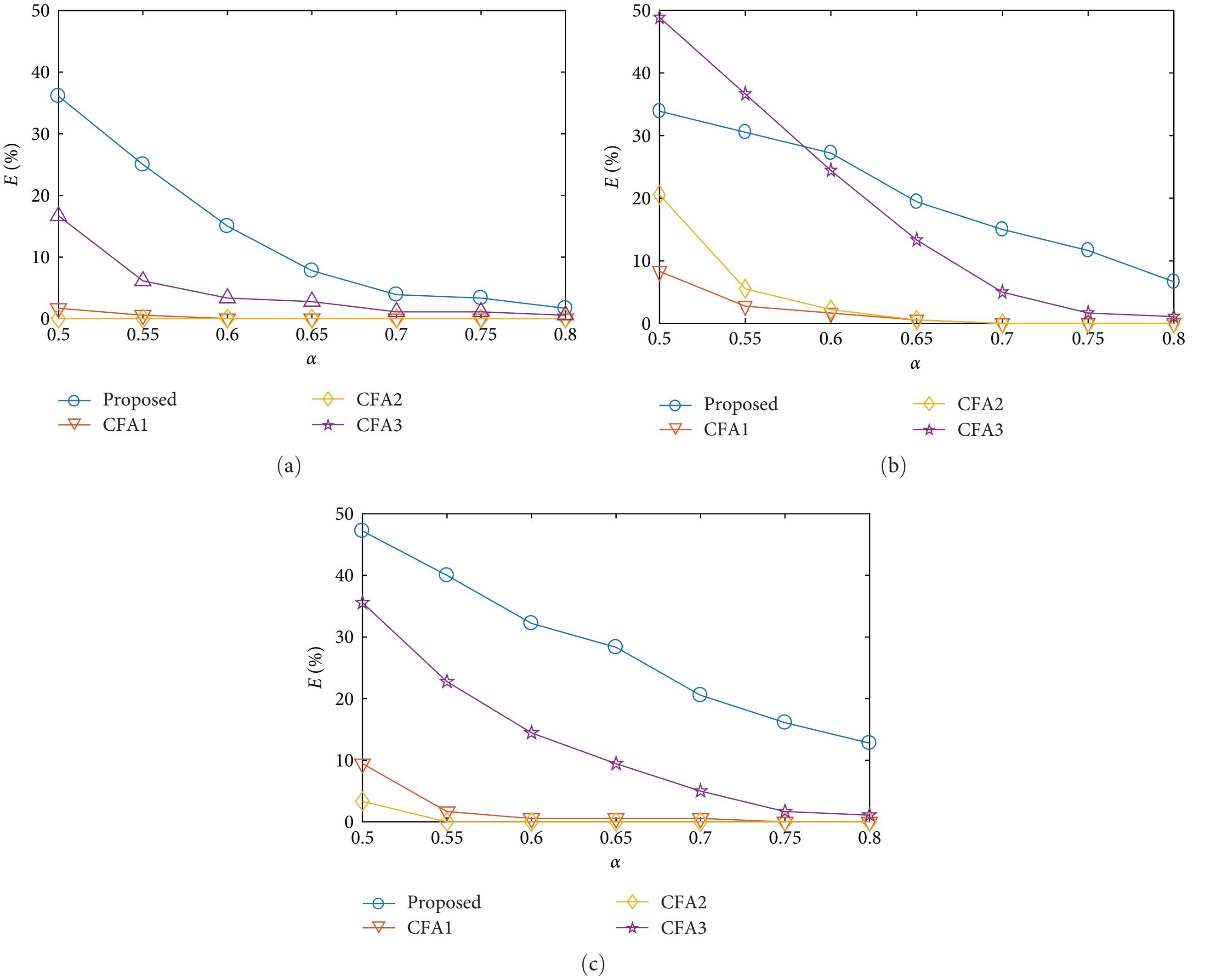
Additionally, we can readily observe that the CFA-based methods perform better than the other four methods. Experiments in Popescu and Farid's [10] study show that the CFA-based methods perform particularly well on the Korus dataset, similar to our experimental results. In fact, experiments in Ferrara et al.'s [27] study show that the CFA-based method has low false positive rate, with a 0% false positive rate in its simulate tampering, which is an important advantage of CFA-based methods. The images of the two datasets used in our experiments are in uncompressed TIFF format, which perfectly preserves the CFA fingerprints. Therefore, the advantages of CFA-based methods are clearly exhibits, making them outperform other methods.

To visually compare the performance of the different methods, Figure 6 shows an example heat map of the localization results. Overall, CFA1, CFA3, and the proposed method outperform the other three methods in locating tampered regions. In the first and second rows, the output of CFA1 presents some false alarms that degrade the performance of the results. Although the forgery localization of CFA3 is rough, the few false alarms make its result scores higher than that of CFA1. The CFA1 method detects detailed parts of the tampered region, but there are many false alarms. The CFA3 has few false alarms, but the results are coarse, and detail is seriously lost. The proposed method detects the details of tampered regions with few false detections.

4.4. Robust Analysis. The experiments in the previous section have demonstrated the robustness of the proposed method to complex scenarios. Subsequently, we test the robustness of the CFA-based methods against various attacks. Since many postprocessing of the whole image completely destroy the CFA fingerprints, we consider only three attacks, namely noise addition, Gaussian filtering, and JPEG compression.

Compared to the RTD dataset, the image resolution in the Columbia dataset is lower. Therefore, this subsection uses the Columbia dataset for the experiments to reduce the computational cost. Three new datasets were generated by attacking the Columbia dataset. (1) We added the familiar Gaussian noise (20 dB) to images to obtain the noise addition dataset. (2) The filtering operation is similar to the interpolation process, and most of the filtering will destroy the CFA fingerprints, such as median filtering and mean filtering. The Gaussian filtering dataset is obtained by Gaussian filtering with filter size of 3 and standard deviation of 0.29. (3) Ferrara et al. [27] tested the sensitivity of their CFA-based method to JPEG compression, the performance quickly drops when the quality is less than 90. Therefore, we use the "imwrite" function in MATLAB to obtain the JPEG compressed dataset with a quality factor of 90.

Figure 7 illustrates the efficiency ratio E under various attacks. Obviously, the proposed method outperforms other CFA-based methods under noise addition and JPEG compression attacks. Figure 6 illustrates that the proposed method gives fine localization results. Therefore, the proposed method provides a high MIoU score, but it also sensitive to noise in the extracted feature. For the images after Gaussian filtering attack, the proposed method results in many high and low



(a)

(b)

(c)

Figure 7: Comparison of efficiency ratio E with Columbia dataset under various attacks: (a) noise addition, (b) Gaussian filtering, and (c) JPEG compression.

Table 2: Average MIoU for Columbia dataset under various attacks.

	Noise addition	Gaussian filtering	JPEG compression
Proposed	0.4978	0.6081	0.6102
CFA1	0.3958	0.4319	0.4292
CFA2	0.404	0.4442	0.4137
CFA3	0.4432	0.5125	0.49

MIoU scores and a little of intermediate scores, i.e., high E scores when α is greater than 0.6 and low E scores when α is greater than 0.5 and less than 0.6. CFA3 gets coarse forgery localization results, thus it is less sensitive to noise in the extracted features. That is, although it gets a little of high MIoU scores, it get a lot of intermediate scores. Therefore, CFA3 has a high E score when α is greater than 0.5 and less than 0.6, but the E score decreases rapidly when α is greater than 0.6. For a more intuitive display, Table 2 shows results with respect to averageMIoU score for the Columbia dataset under various attacks. In all three new datasets, the proposed method ranks first. Moreover, it is 12.31%, 18.65%, and 24.53% better than the second-best method (CFA3), which is much larger than 1.55% on the Columbia dataset. Although the performance of the CFA-based method is significantly degraded under various attacks, the proposed method has more significant advantages over other CFA-based methods.

5. Conclusion

In this paper, we propose a CFA-based forgery localization method. Most previous CFA-based methods assumed the interpolation algorithm is linear, which is impractical for commercial cameras. In contrast, the proposed method is based on the fact that an interpolated pixel value falls in the range of its neighboring acquired pixel values, which is valid for both linear and nonlinear interpolation algorithms. The proposed method outperforms the reference methods and is more robust to the tested attacks.

The CFA-based forgery localization method mainly considers raw images. Although these images are rarely present in daily life, they still exist in certain fields, such as copyright protection. For raw images, the CFA-based method has a low false detection rate and outperforms most methods. Therefore, the CFA-based forgery localization methods are still useful tools in practical applications. In the future, we will try to combine the CFA-based method with various other methods to make them applicable for practical applications.

Data Availability

The databases used to support the findings of this study are included within the article [33, 34]. The codes used to support the findings of this study are included within the article [40].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Technical Research Program of Ministry of Public Security (2020JSYJC25) and the Open Project of Key Laboratory of Forensic Science of Ministry of Justice (KF202317).

References

[1] D. Cozzolino and L. Verdoliva, "Noiseprint: a CNN-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.

[2] H. Li, W. Luo, X. Qiu, and J. Huang, "Image forgery localization via integrating tampering possibility maps," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1240–1252, 2017.

[3] B. Peng, W. Wang, J. Dong, and T. Tan, "Optimized 3D lighting environment estimation for image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 479–494, 2017.

[4] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. de Rezende Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, 2013.

[5] E. Kee, J. F. O'brien, and H. Farid, "Exposing photo manipulation from shading and shadows," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 1–21, 2014.

[6] F. Matern, C. Riess, and M. Stamminger, "Gradient-based illumination description for image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1303–1317, 2020.

[7] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6231–6235, IEEE, Florence, Italy, 2014.

[8] H. Yao, S. Wang, X. Zhang, C. Qin, and J. Wang, "Detecting image splicing based on noise level inconsistency," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12457–12479, 2017.

[9] I. Yerushalmy and H. Hel-Or, "Digital image forgery detection based on lens and sensor aberration," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 71–91, 2011.

[10] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[11] Q. Bammey, R. Grompone von Gioi, and J.-M. Morel, "An adaptive neural network for unsupervised mosaic consistency analysis in image forensics," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14194–14204, IEEE, 2020.

[12] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensic detection of resampled images," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 536–545, 2012.

[13] J. Bunk, J. H. Bappy, T. M. Mohammed et al., "Detection and localization of image forgeries using resampling features and deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1881–1889, IEEE, Honolulu, HI, USA, 2017.

[14] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: a difference domain based approach," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4699–4710, 2013.

[15] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

[16] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 515–525, 2014.

[17] J.-Y. Sun, S.-W. Kim, S.-W. Lee, and S.-J. Ko, "A novel contrast enhancement forensics based on convolutional neural networks," *Signal Processing: Image Communication*, vol. 63, pp. 149–160, 2018.

[18] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on Dempster–Shafer theory of evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, 2013.

[19] B. G. Jeong, Y. H. Moon, and I. K. Eom, "Blind identification of image manipulation type using mixed statistical moments," *Journal of Electronic Imaging*, vol. 24, no. 1, Article ID 013029, 2015.

[20] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 5297–5301, IEEE, Paris, France, 2014.

[21] J. J. Jeon, H. J. Shin, and I. K. Eom, "Estimation of Bayer CFA pattern configuration based on singular value decomposition," *EURASIP Journal on Image and Video Processing*, vol. 2017, Article ID 47, 2017.

[22] Q. Bammey, R. Gioi, and J.-M. Morel, "Automatic detection of demosaicing image artifacts and its use in tampering detection," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, Miami, FL, USA, 2018.

[23] E. G. Fernández, A. S. Orozco, L. G. Villalba, and J. Hernandez-Castro, "Digital image tamper detection technique based on spectrum analysis of CFA artifacts," *Sensors*, vol. 18, no. 9, Article ID 2804, 2018.

[24] C.-H. Choi, J.-H. Choi, and H.-K. Lee, "CFA pattern identification of digital cameras using intermediate value counting," in *Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security*, pp. 21–26, ACM, 2011.

[25] C.-H. Choi, H.-Y. Lee, and H.-K. Lee, “Estimation of color modification in digital images by CFA pattern change,” *Forensic Science International*, vol. 226, no. 1–3, pp. 94–105, 2013.

[26] H. J. Shin, J. J. Jeon, and I. K. Eom, “Color filter array pattern identification using variance of color difference image,” *Journal of Electronic Imaging*, vol. 26, no. 4, Article ID 043015, 2017.

[27] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, “Image forgery localization via fine-grained analysis of CFA artifacts,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[28] A. Singh, G. Singh, and K. Singh, “A Markov based image forgery detection approach by analyzing CFA artifacts,” *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28949–28968, 2018.

[29] L. Lu, R. Ni, Z. Yao, and S. Li, “Improved SIFT-based copy-move detection using bfsn clustering and CFA features,” in *Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 626–629, IEEE, Kitakyushu, Japan, 2014.

[30] T.-Y. Chang, S.-C. Tai, and G.-S. Lin, “A passive multi-purpose scheme based on periodicity analysis of CFA artifacts for image forensics,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 6, pp. 1289–1298, 2014.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[32] D. Reynolds, “Gaussian mixture models,” in *Encyclopedia of Biometrics*, pp. 659–663, Springer, Boston, MA, 2009.

[33] Y. F. Hsu and S. F. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency,” in *2006 IEEE International Conference on Multimedia and Expo*, pp. 549–552, IEEE, Toronto, ON, Canada, 2006.

[34] P. Korus and J. Huang, “Multi-scale analysis strategies in PRNU-based tampering localization,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, 2017.

[35] A. E. Dirik and N. Memon, “Image tamper detection based on demosaicing artifacts,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 1497–1500, IEEE, Cairo, Egypt, 2009.

[36] W. Li, Y. Yuan, and N. Yu, “Passive detection of doctored JPEG image via block artifact grid extraction,” *Signal Processing*, vol. 89, no. 9, pp. 1821–1829, 2009.

[37] C. Iakovidou, M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, “Content-aware detection of JPEG grid inconsistencies for intuitive image forensics,” *Journal of Visual Communication and Image Representation*, vol. 54, pp. 155–170, 2018.

[38] B. Mahdian and S. Saic, “Using noise inconsistencies for blind image forensics,” *Image and Vision Computing*, vol. 27, no. 10, pp. 1497–1503, 2009.

[39] H. Zeng, Y. Zhan, X. Kang, and X. Lin, “Image splicing localization using PCA-based noise level estimation,” *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 4783–4799, 2017.

[40] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, “Large-scale evaluation of splicing localization algorithms for web images,” *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 4801–4834, 2017.

[41] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” 2017.