Forensic Science International 307 (2020) 110109

Contents lists available at ScienceDirect

Forensic Science International

journal homepage: www.elsevier.com/locate/forsciint

A unified framework of source camera identification based on features

Bo Wang[a], Kun Zhong[a,*], Zihao Shan[b], Mei Neng Zhu[c], Xue Sui[d]

[a] School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, Liaoning, PR China
[b] Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY 14260-2500, USA
[c] Beijing Institute Of Electronics Technology And Application, Beijing 100091, PR China
[d] College of Psychology, Liaoning Normal University, Dalian, Liaoning 116029, PR China

ARTICLE INFO

Article history:
Received 26 August 2019
Received in revised form 6 December 2019
Accepted 9 December 2019
Available online 19 December 2019

Keywords:
Source camera identification
Image features
Framework

ABSTRACT

Source camera identification, which aims at identifying the source camera of an image, has attracted a wide range of attention in the field of digital image forensics recently. Many approaches to source camera identification have been proposed by extracting some image features. However, most of these methods only focused on extracting features from the single artifact of the camera left on the captured images and ignored other artifacts that may help improve final accuracy. Therefore, in this paper, we propose a feature-based framework for source camera identification, which first captures various pure camera-specific artifacts through preprocessing and residual calculation, then extracts discriminative features through image transform, and finally reduces the algorithm complexity through feature reduction. Based on the framework, a novel source camera identification method is proposed, which can identify different camera brands, models and individuals with high accuracy. A large number of comparative experiments show that the proposed method outperforms the state-of-the-art methods.

1. Introduction

With the rapid development of the Internet and the popularity of digital cameras and smartphones, digital images have gradually become the primary way for people to record and disseminate information. Accordingly, a large number of practical and easy-to-operate image processing software applications emerge constantly. Though providing us with much convenience, these applications also incur notable security concerns regarding the content of digital images, especially in the field of judicial. Consequently, reliable image forensics technologies are urgently needed to verify the originality, authenticity, and reliability of given images.

As one of the most important branches of digital image forensics, source camera identification (SCI) aims to identify the brand, type or individual of the camera used to capture a given image. Most of the existing SCI methods belong to passive forensics, i.e., these methods only utilize the images taken by camera without access to any camera-related information. These methods are feasible because of the complicated imaging pipeline of the camera, which inevitably leaves some unique traces on the captured images. A general camera imaging process is illustrated in Fig. 1. The reflected light from the natural scene enters the camera through the lens and then passes through some filters. After that, the light will be focused on the most important component of the camera—sensor, which can then transform an optical signal into an electrical signal.

To reduce costs and improve efficiency, most manufacturers equip the color filter array (CFA) on the top of the sensor, ensuring that only one color is allowed at each pixel. Then, CFA interpolation is performed to convert the demosaicing image into a true color image, and other post-processing algorithms such as gamma correction, white balance, and JPEG compression are performed to improve image quality. Different brands or models of cameras will leave the camera-specific artifacts on the captured images with respect to the different hardware and signal processing algorithms. According to different means of artifact processing, the existing source camera identification methods can be roughly classified into two categories: correlation-based methods and feature-based methods.

Due to the limitations of technology and materials, most camera hardware has some defects, such as optical defects, sensor defects and so on, which will leave unique hardware-related artifacts on the captured images. Among these artifacts, the most widely used is sensor pattern noise (SPN), which is also the key to correlation-based methods. It is worth noting that

* Corresponding author.
E-mail addresses: bowang@dlut.edu.cn (B. Wang), 898782924@qq.com (K. Zhong), zihaosha@buffalo.edu (Z. Shan), zmneng@163.com (M.N. Zhu), suixue@lnnu.edu.cn (X. Sui).

http://dx.doi.org/10.1016/j.forsciint.2019.110109
0379-0738/

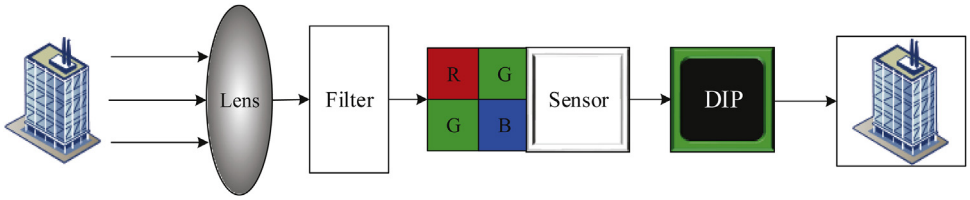

Fig. 1. Imaging pipeline of the camera.

since photo-response non-uniformity (PRNU) noise is the main component of SPN, they are considered equivalent in this paper. Lukas et al. [1] creatively used PRNU for source camera identification in 2006. The SPN of each image was obtained by wavelet filtering proposed in [2], and then the camera reference SPN was obtained by averaging the SPN of multiple images captured by the same camera. The camera whose RPN has the highest correlation with the SPN of the test image is considered as the corresponding source camera. Based on that, researchers have proposed many improved methods, mainly focusing on three aspects: SPN extraction, SPN enhancement, and correlation calculation. Since the extraction of SPN is achieved by denoising through a filter, the optimal denoising filter selection is critical for source camera identification. Inspired by the camera sensor output model, Chen et al. [3] proposed the maximum likelihood estimation (MLE) method to extract PRNU. Wu et al. [4] proposed an SPN predictor based on content adaptive interpolation (PCAI), in which four adjacent pixels are used to interpolate the central pixel. Kang et al. [5] extended the PCAI method by interpolating with eight adjacent pixels instead of four adjacent pixels. Recently, He et al. [6] proposed a novel image filter called guided image filter, whose strong edge-preserving properties make it suitable for realistic estimation of image scene content. The guided image filter has been used for SPN extraction by Zeng et al. in [7]. Although various filters have been applied to the extraction of SPN, the extracted SPN is still contaminated by image content, and non-unique artifacts (NUA) shared between different cameras. Therefore, how to enhance the extracted SPN is one of the research hotspots. In [3], two preprocessing steps, zero mean (ZM) and Wiener Filter (WF), were proposed to suppress the unwanted artifacts in extracted SPN. However, the proposed operations cannot effectively reduce the impact of image content. Based on the hypothesis that the component in SPN with large magnitude is more likely to be contaminated by the image content, Li [8] proposed five enhancing models to attenuate the interference from image content. Kang et al. [9] considered this problem in the frequency domain and calculated the reference phase SPN by averaging the phase component of the SPN spectrum to remove the impact of image content and NUA. Also, in the frequency domain, Lin et al. [10] proposed a spectrum equalization algorithm (SEA) to remove the periodic interference from RPN. Besides, some weighted average (WA) methods are proposed to construct RPN with less image scene details [11,12]. When calculating the correlation between SPN and RPN, normalized cross-correlation (NCC) detector was firstly proposed by Lukas et al. [1]. Subsequently, the peak-to-correlation-energy ratio (PCE) and circular cross-correlation norm (CNN) were successively applied to improve the receiver operating characteristic (ROC) curve and reduce the computational cost.

Although correlation-based methods have made significant progress in recent years, they are still limited by the purity of the SPN and the high complexity of the matching process. With the development of machine learning, some researchers have proposed the feature-based methods for source camera identification. Kharrazi et al. [13] fed three sets of basic statistical features into the support vector machine (SVM) classifier to perform source camera identification. On this basis, some extended basic statistical features were proposed to enhance the detection accuracy [14,15]. Wang et al. [16] extracted the higher-order wavelet statistics (HOWS), wavelet coefficient co-occurrence features from the images and selected the optimal features through sequential forward feature selection (SFFS) algorithm to train the classification model. Celiktutan et al. [17] combined binary similarity metrics (BSM) with HOWS for source camera identification. Xu et al. [18] used uniform gray-scale invariant local binary patterns (LBP) to capture the difference in image texture caused by various camera hardware and image processing algorithms. In addition to spatial and wavelet domain features, some features extracted from the frequency domain, such as conditional probability (CP) features [19,20] and discrete cosine transform residue (DCTR) features [21], have also been applied to source camera identification. In general, the feature-based methods mentioned above captured the global variations in the imaging process of different cameras by extracting the first-order or higher-order statistical features. Other methods focus on specific artifacts left behind by different in-camera image processing algorithms. Wang et al. [22] proposed a novel method based on CFA interpolation, in which, 1022 CFA interpolation coefficients were estimated as the features by using covariance matrix to improve the robustness to JPEG compression. Although the methods based on parameter estimation are effective, the estimation processes for parameters are generally complicated. Instead of attempting to estimate the parametric models of the image processing algorithms, Chen et al. [23] proposed a novel framework for camera model identification by using the co-occurrence matrix features extracted from the rich model of the CFA interpolation algorithms. It is worth noting that the ensemble classifier was used in [21,23] instead of SVM to deal with high-dimensional features for computational efficiency. Rather than focusing on CFA interpolation, Deng et al. [24] proposed a method based on automatic white balance (AWB) for source camera identification. In [24], six different AWB algorithms were used to re-balance the original image, and then the IQM features extracted from the six re-balanced images were sent to the SVM for classification. It is noteworthy that the feature-based methods mentioned above are generally applicable to the identification of camera models because of the similarity of the image processing algorithms. In recent years, researchers have found that the features extracted from SPN can identify the different camera individuals. Accordingly, some methods based on SPN features have been proposed. Akshatha et al. [25] performed the three-level wavelet decomposition on the SPN through a quadrature mirror filter (QMF) and then computed the statistical features for the high-frequency subband coefficients of each level. Xu et al. [26] extracted LBP and local phase quantization (LPQ) features from the original image, SPN and the contourlet transform coefficients of SPN respectively, and combined them into a feature set to improve the accuracy of the proposed method for camera individuals identification.

It can be seen that features extracted from camera-specific artifacts often perform better than those extracted directly from

the original image. However, most existing feature-based methods focused on finding new features available or merging multiple features extracted from single camera-specific artifact, which fail to make full use of the camera legacy information. In this paper, we propose a general feature-based source camera identification framework, which can effectively capture various camera-specific artifacts and extract discriminative features from these artifacts. The proposed framework consists of six parts: image preprocessing, residual image calculation, image transform, feature extraction, feature dimension reduction, and classification. Preprocessing and residual image calculation can capture camera hardware-related artifacts and software-related artifacts, in which the image content is suppressed as much as possible. Image transform obtains more information about these artifacts from different domains, making the extracted features more discriminative. Feature dimension reduction can reduce redundancy and avoid overfitting. Based on this framework, a novel feature-based method is proposed for source camera identification, which can identify different camera brands, models and individuals with high accuracy. The main contributions of our work are as follows:

- We summarize the existing feature-based methods and propose a unified feature-based source camera identification framework, which can provide the basis for the future research on feature-based SCI.
- We explore various algorithm options in different parts of the proposed framework and propose a novel source camera identification method that performs best in experiments. The proposed method carries out demosaicing, predicting and de-noising algorithms in preprocessing module, performs a wavelet transform in image transform module, combines statistical moment features with LBP features in feature extraction module, and uses Liblinear in classification module.
- Different from most existing methods that only train one classification model, we first train a brand model to obtain the camera brand, then find the type model corresponding to this brand to detect the camera model, and finally use the corresponding individual model to detect the source camera individual. The hierarchical experimental setup improves the final detection accuracy by 1.5%.

The rest of this paper is organized as follows: in Section 2, we describe the proposed framework and enumerate some existing algorithms for each module of the framework. Section 3 shows the experimental results when different algorithms are used in each module of the proposed framework. Finally, the conclusions are drawn in Section 4.

2. Proposed framework

The motivation for the proposed method is that we want to capture as much as possible the discriminative information left by the camera on the image and present them in the form of features. So for that, a feature-based framework is proposed, which consists of six parts: image preprocessing, residual image calculation, image transform, feature extraction, feature dimension reduction, and classification. The overview of the proposed framework is shown in Fig. 2. First, we use some preprocessing algorithms to process the original image separately, which will produce a series of pre-processed images. Then we subtract the pre-processed images from the original image to obtain the residual images, whose main components are the traces left by the camera's hardware characteristics or software characteristics. The process of residual image calculation is important because it effectively suppresses the influence of image content on source camera identification. Next, the image transform is performed on each residual image to get more information from different domains. To avoid the loss of information, we also perform the image transform on the original image. Finally, some statistical features are extracted from these transformed residual images and the original image. It can be seen that to capture and represent the camera-specific characteristics comprehensively, the combination of multiple algorithms in each module of the framework is inevitable, which will produce high-dimensional features. Therefore, to reduce the algorithm complexity and avoid over-fitting, it is necessary to reduce the dimension of the obtained features before training the classification model. Next, we will detail some existing algorithms for each module of the framework.

2.1. Image preprocessing

Image preprocessing is the core of this framework and is also the basis of residual image calculation in the next step. In this section, four preprocessing algorithms are considered, namely de-noising, demosaicing, re-balancing, and predicting. De-noising is used to extract the SPN noise of the image. Demosaicing and re-balancing can capture information about the image processing algorithms by generating some sub-images. Predicting can explore the impact of the camera imaging process on image smoothness. In the following, the four preprocessing algorithms will be described in detail.

2.1.1. De-noising

As mentioned above, defects in the camera's internal hardware will leave unique artifacts on the captured images, which can be used to distinguish different camera individuals of the same model. Among these hardware-related artifacts, the noise caused by sensor imperfections, SPN, is the most widely used in recent years. In most of the existing SPN-based methods, the output of the imaging sensor is modeled as:

$$I = I_0 + KI_0 + N_r \tag{1}$$

where I is the actual sensor output, I_0 is the noiseless sensor output and N_r represents the additive random noise. KI_0 is the signal of our interest, that is, SPN noise. Unfortunately, it is difficult to extract the SPN directly from the actual output I. Instead, we can first estimate the noiseless output I_0 by de-noising the actual sensor output I, i.e., $\hat{I}_0 = F(I)$, where F represents the de-noising algorithm and $\hat{I}_0$ is an approximate estimation of I_0. Then, the estimated SPN can be obtained by subtracting I_0 from I.

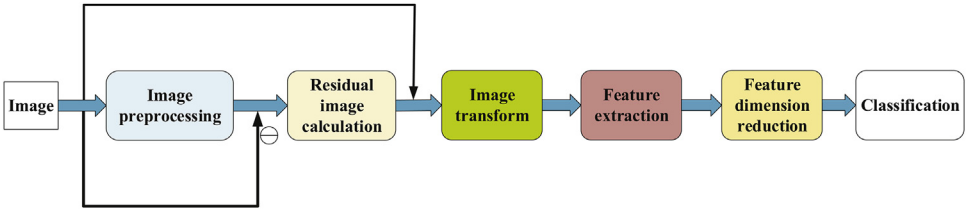

Fig. 2. Flowchart of our proposed source camera identification framework.

When identifying the source camera, the quality of the extracted SPN directly affects the final detection accuracy. The estimated SPN, however, is often contaminated by the image content I_0 and the random noise N_r, resulting in a decrease in the detection accuracy. In order to get a pure SPN, it is essential to select a suitable de-noising filter. In this paper, the wavelet-based de-noising filter script provided by Binghamton University [1] is used to de-noise the image.

2.1.2. Demosaicing

Considering cost efficiency and robustness, most commercial cameras add a CFA in front of the sensor to record the three color components of light (R, G, and B) using one sensor. The most common CFA pattern is Bayer, as shown in Fig. 3.

Although only one color value is observed at each pixel, the remaining two color values can be interpolated by using a process called demosaicing. According to whether the correlation between different color channels is utilized, existing demosaicing algorithms can be divided into two categories: correlation and non-correlation. The non-correlation based methods utilize observations from a single color channel to estimate the remaining unobserved pixel values for that channel, while the correlation based methods take into account the relationship between different color channels when performing pixel interpolation. The diversity of demosaicing algorithms gives camera manufacturers more choices, making the demosaicing algorithms become one of the camera-specific characteristics. Based on this, various source camera identification methods were proposed by building the demosaicing algorithm model and estimating model parameters. However, as mentioned above, the estimation of algorithm parameters is complex and inaccurate. To avoid estimating model parameters, we use a rich model of demosaicing algorithms proposed in [23] to capture information about different demosaicing algorithms. As is shown in Fig. 4, the original image is first re-sampled by CFA to obtain the reconstructed image, which is an approximate estimation of the camera internal image before CFA interpolation. Then, various demosaicing algorithms are performed on the reconstructed image to obtain multiple output images. If the demosaicing algorithms used twice are the same, the output image is similar to the original image. Otherwise, the two images are different. Therefore, features extracted from the output images can be used to identify the source camera. In this paper, the Bayer pattern is used to re-sample the original image and two common algorithms: nearest neighbor and bilinear interpolation are used for demosaicing.

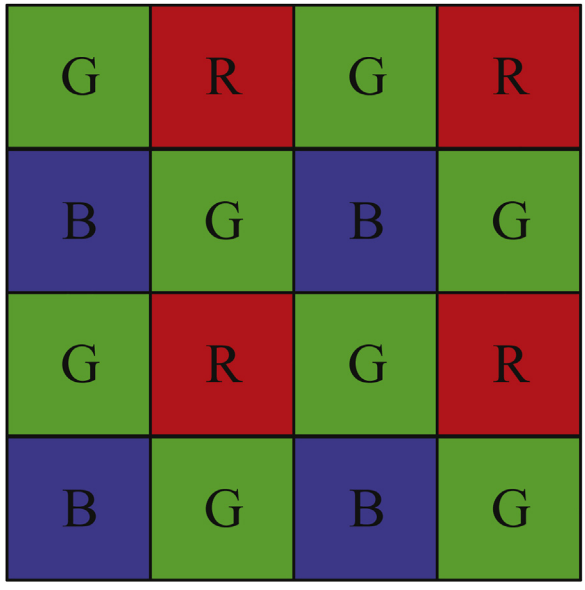

Fig. 3. The Bayer pattern. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

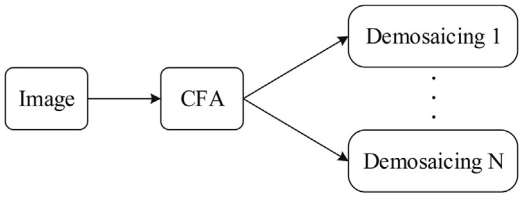

Fig. 4. The process of demosaicing.

2.1.3. Re-balancing

White balance is widely used in digital cameras to correct the color deviation caused by the light source between the captured image and the actual scene. The white balance algorithms of different camera models generally differ from each other, which can be used to identify the source camera. In order to distinguish cameras by capturing the difference between the white balance algorithms, the straightforward idea is to estimate the white balance parameters from the captured image as features for classification. However, due to the diversity of white balance algorithms, the estimation of their parameters is complicated and inaccurate. Instead of estimating the parameters, Deng et al. [24] utilized various kinds of white balance algorithms to re-balance the original image and then extract features from the re-balanced images. The core idea of the method is that since the color deviation has been removed by the first white balance, the second white balance has little effect on the original image when the algorithm used is the same as the first white balance, i.e., given an image I, the following formula is established:

$$WB^i(WB^i(I)) = WB^i(I) \tag{2}$$

where WB^i represents white balance using the algorithm i. However, when the algorithms used twice are different, the re-balance operation will change the original image in some way. Therefore, the re-balanced images can reflect the white balance algorithm used in the original image, which corresponds to the camera model. It is reasonable that the performance of the algorithm will be improved when using more white balance algorithms. Considering the trade-off between complexity and accuracy, we execute the following four white balance algorithms in preprocessing:

1. Gray-World
2. White-Patch
3. Shades-of-gray
4. Gray-Edge (with differentiation order 1 and 2)

2.1.4. Predicting

In general, it is thought that natural images are smooth in weak texture regions, that is, a pixel value can be accurately predicted by its neighboring pixel values. However, noise and image processing algorithms during camera imaging can affect the smoothness of the image, resulting in a difference between the predicted image and the real image. These differences related to camera-specific characteristics can be applied to source camera identification. In preprocessing, we use eight neighboring pixels to linearly predict the central pixel [27], as shown in Fig. 5.

Given an image I, we first identify its weak texture regions based on the local gradient. For each pixel of the image, the

b_{i1}	b_{i2}	b_{i3}
b_{i4}	a_i	b_{i5}
b_{i6}	b_{i7}	b_{i8}

Fig. 5. Eight neighboring pixels of a_i.

-1	0	1
-1	0	1
-1	0	1

Horizontal

-1	-1	-1
0	0	0
1	1	1

Vertical

Fig. 6. Prewitt operators.

horizontal gradient g_h and the vertical gradient g_v can be calculated by using the Prewitt operators shown below (Fig. 6):

Pixels with a gradient less than the threshold g are considered as weak texture regions. In this paper, we set the threshold $g = 10$. For each pixel a_i in the weak texture regions, the predicted value $\hat{a}_i$ can be obtained by the following formula:

$$\hat{a}_i = \sum_{k=1}^{8} w_k b_{ik} \tag{3}$$

where $w_k \geq 0$ is the weight corresponding to the kth neighboring pixel of a_i. For clarity, we denote the weight w_k as a column vector $W = [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8]^T$ and represent the N pixel values of the weak texture regions as a column vector $A = [a_1, a_2, \ldots, a_N]$. The neighboring pixels of the N pixels can be represented by a matrix $B = [b_1, b_2, \ldots, b_N]^T$ in which $b_i = [b_{i1}, b_{i2}, \ldots, b_{i8}]$ represents the eight neighboring pixels of a_i. Then the weight w_k can be calculated by solving a non-negative least squares problem:

$$\min_x \quad BW - A^2, s.t. \mathrm{W} \geq 0 \tag{4}$$

Finally, a predicted image will be obtained by calculating the predicted value of each pixel in the weak texture regions using formula (3).

2.2. Residual image calculation

After preprocessing, the corresponding outputs of the four preprocessing algorithms are one de-noised image, two demosaiced images, five re-balanced images and one predicted image respectively. As mentioned, the purpose of de-noising is to estimate the SPN indirectly, so the residual noise image I_{SPN} whose main component is SPN can be obtained by subtracting the de-noised image I_0 from the original image $\hat{I}_0$.

$$I_{SPN} = I_0 - \hat{I}_0 \tag{5}$$

Unlike de-noising, the output images obtained by other preprocessing algorithms in this paper contain camera-specific characteristics and can be directly used for source camera identification. However, the features extracted from these output images are often affected by the image content information, resulting in a reduction in detection accuracy. To focus on camera artifacts, we subtract the output images from the original image to obtain the residual images, which only record the errors caused by the different image processing algorithms. Therefore, the residual images can capture more pure camera artifacts and extract more discriminative features than the preprocessing output image. In Fig. 7, we take demosaicing as an example to show the effectiveness of residual images visually. LBP features are extracted from the demosaicing output images and the demosaicing residual images, respectively, and projected into 3-dimensional space by principal component analysis (PCA). It can be seen that the distance between the features extracted from the residual images

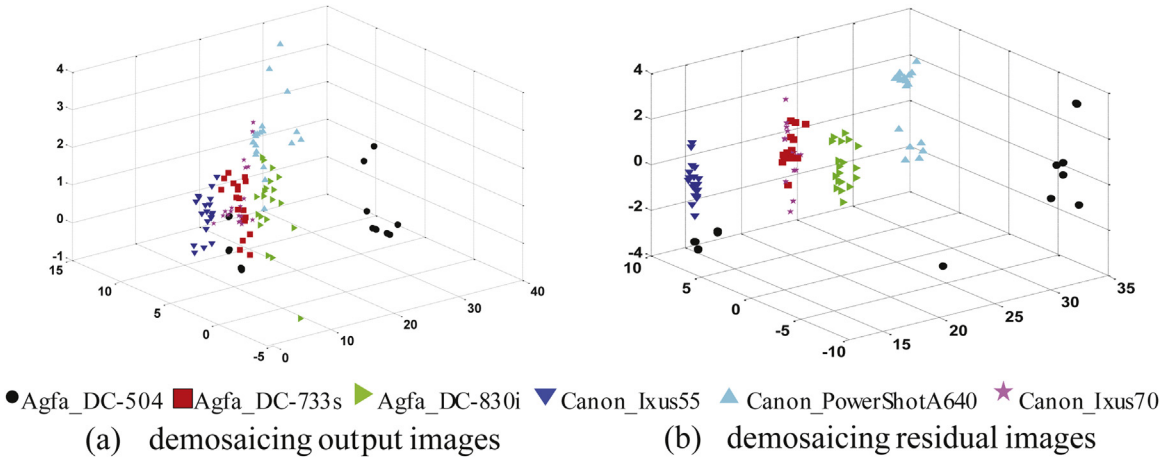

Fig. 7. 3-D projection results from the whole feature set by PCA. Symbols of different colors and shapes indicate different cameras. (a) LBP features are extracted from the demosaicing output images (b) LBP features are extracted from the demosaicing residual images. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

of different cameras is larger, meaning that the features extracted from the residual images are more discriminative.

2.3. Image transform

In signal processing, some problems can be easily solved by converting signals from the time domain to other domains. Similarly, we believe that the features extracted from the image transform domain can perform better in source camera identification than the spatial features, which will be verified in the experimental section. In this section, we consider two image transform algorithms that are widely used in source camera identification: wavelet transform and contourlet transform.

- A three-level wavelet transform using Haar wavelet is performed on the original image and residual images. Each level of wavelet decomposition produces a low-frequency subband LL and three high-frequency subbands: HL (horizontal), LH (vertical) and HH (diagonal). The second level wavelet decomposition is performed on the LL subband generated by the first level decomposition. For R, G and B channel of an image, the above decomposition is performed respectively, which result in 27 high-frequency subbands.
- Compared with the wavelet transform, the contourlet transform is more directional due to its special filter bank structure. In this filter bank structure, the Laplacian pyramid (LP) is used to find the point singularities, and a directional filter bank (DFB) is used to combine these singularities to linear structures. The combination of LP and DFB can decompose the image in multiple directions. In this paper, a three-level contourlet transform is performed, and four sub-bands are taken in each level decomposition. This process is performed on the R, G, and B channels of an image, respectively, resulting in 36 high-frequency subbands.

2.4. Feature extraction

In recent years, various features have been applied to source camera identification, such as LBP features, LPQ features, co-occurrence features, CP features, etc. In theory, these features are all applicable to the proposed framework. However, in the previous sections, various preprocessing algorithms are used to capture more camera artifacts, resulting in multiple residual images. Then, the image transform is performed on each residual image to obtain the high-frequency subbands. In this paper, the features are extracted from all high-frequency coefficient matrixes, which is time-consuming. In this case, it is preferable to select simple and low dimensional features, as shown below.

IQM features: Image Quality Metric is one of the most important indicators for quantitative evaluation of image quality. The 13-dimensional IQM features were first used for source camera identification by Kharrazi et al. in [13]. The low dimensionality and computational simplicity make the features suitable for the proposed framework. According to the different measurement methods, IQM features can be divided into three categories.

1. Pixel difference based measures: mean absolute error (MAE), mean square error (MSE), modified infinity norm.
2. Correlation based measures: cross correlation, Czekonowski correlation.
3. Spectral based measures: magnitude error, spectral phase.

Statistical moment features: In statistics, moments are the representation of the distribution of a random variable. There are many types of moments, each of which reflects some distribution information of random variable. For example, the first moment, second moment, third moment and fourth moment represent the mean, variance, skewness and kurtosis of the variable, respectively. In this paper, these four statistical moments are calculated as features by the following formula:

$$Mean(X) = E(X) \tag{6}$$

$$\mathrm{Var}(X) = E\{[X - E(X)]^2\} \tag{7}$$

$$Skew(X) = E\left\{\left[\frac{X - E(X)}{\sqrt{\mathrm{Var}(X)}}\right]^3\right\} \tag{8}$$

$$Kurt(X) = E\left\{\left[\frac{X - E(X)}{\sqrt{\mathrm{Var}(X)}}\right]^4\right\} \tag{9}$$

where X is a random variable, $E(\cdot)$ is a mean function.

LBP features: LBP operator was first proposed by Ojala et al. [28] to reflect the local texture information of an image. Due to its computational simplicity and high accuracy, LBP operator has become one of the most widely used methods in many fields. Xu et al. [18] used them as features for source camera identification and achieved better performance. Taking a $3*3$ window as an example, the original LBP is defined as shown in Fig. 8. The center pixel value is taken as a threshold and compared with the eight neighboring pixel values. When the neighbor pixel value is greater than the threshold, it is encoded as 1, otherwise 0. Finally, each pixel in the image will correspond to an 8-bit binary number, called LBP value. By calculating the histogram of these LBP values, a $2^8 = 256$ dimensional feature can be obtained. Later, Ojala et al. proposed the uniform LBP based on the original LBP. The uniform LBP reduces the feature dimension from 256 to 59 according to the number of binary transitions within the LBP value.

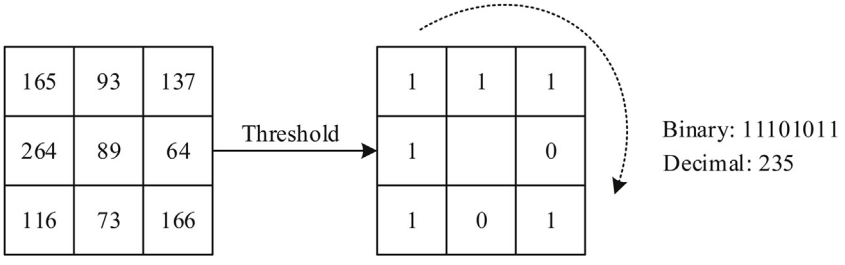

Fig. 8. The encoding process of LBP.

2.5. Feature dimension reduction

Feature dimension reduction generates new low-dimensional features from the original high-dimensional features through feature projection to avoid over-fitting and reduce redundant information. Feature projection not only reduces the dimension but also eliminates the correlation between features and discovers some potentially useful features. Therefore, a classic feature dimension reduction method, PCA is used in this paper to obtain a low-dimensional projection of the original features. In signal processing, it is generally believed that the signal has a large variance and the noise has a small variance. The core idea of PCA is to find an appropriate matrix to project the feature so that the data variance after dimension reduction is maximized, in which case the principal components of the original data are preserved. To find the optimal matrix, we first center the data by subtracting the mean of all samples from the features. Then, the covariance matrix of the data, the eigenvalues of the covariance matrix and the corresponding eigenvectors are calculated successively. Sorting the eigenvalues in descending order, and the matrix composed of the unit eigenvectors corresponding to the top k eigenvalues is the optimal projection matrix. k is the feature dimension after projection, which can be determined by the following formula:

$$k^* = \min \left\{ k | \sum_{j=1}^{k} D_j \geq \lambda \sum_{j=1}^{m} D_j \right\} \tag{10}$$

where D_j is the jth sorted eigenvalue, m is the number of all eigenvalues and λ represents the percentage of principal component information in total data information.

2.6. Classification

In this section, two mainstream classifiers are considered to classify the features obtained:

SVM classifier: A supervised classification model based on statistical learning theory, which is widely used for feature-based source camera identification due to its good generalization ability. The use of nuclear method enables SVM to solve complex nonlinear classification problems. In this paper, the LIBSVM package [29] with radial basis function (RBF) kernel and Liblinear package [30] are used for experiments. It is worth noting that although the essence of Liblinear is LIBSVM with linear kernel, its optimization algorithm is completely different from LIBSVM, which makes it possible to achieve the same classification results as LIBSVM in the linear classification with lower algorithm complexity.

Ensemble classifier: Since the complexity of SVM will increase greatly with the sample size and feature dimension, ensemble classifier has attracted more and more attention because of its low training complexity. The main idea of ensemble classifier is to use multiple weak classifiers with independent decision-making ability and then make decision fusion. In this paper, an ensemble classifier, random forest (RF) is used for experiments.

3. Experiment results

In order to comprehensively evaluate the performance of the proposed framework, a large number of experiments are carried out in this section:

(1) we perform component analysis to verify the importance of each module in the proposed framework and explore the various algorithm options in different modules;
(2) we compare the proposed method with the state-of-the-art feature-based methods.

Table 1
The parameters of cameras and images.

Camera ID	Make	Model	Number
1	Agfa	DC-504	150
2	Agfa	DC-733s	150
3	Agfa	DC-830i	150
4	Canon	Ixus55	150
5	Canon	Ixus70	150
6	Canon	PowerShotA640	150
7	Casio	EX-Z150	150
8	FujiFilm	FinePixJ50	150
9	Nikon	D200(1)	150
10	Nikon	D200(2)	150
11	Nikon	D70(1)	150
12	Nikon	D70(2)	150
13	Olympus	mju_1050SW	150
14	Panasonic	DMC-FZ50	150
15	Sony	DSC-H50(1)	150
16	Sony	DSC-H50(2)	150
17	Sony	DSC-T77(1)	150
18	Sony	DSC-T77(2)	150
19	Sony	DSC-W170(1)	150
20	Sony	DSC-W170(2)	150

The experimental cameras are all from the Dresden Image Database [31], as shown in Table 1. The selected cameras consist of 20 different camera individuals, including eight camera brands, and each camera brand contains different camera models. Besides, different camera individuals of the same brand and model are also included, such as two different individuals of Nikon_D200 (ID 9, 10). For each camera, 150 full resolution images are used, 100 for training and 50 for testing.

Different from most existing methods that only train one classification model, a hierarchical idea is proposed in this paper. We train three kinds of classification models, namely brand model, type model, and individual model. When identifying the source camera of the test image, we first use the brand model to obtain the camera brand, then find the type model corresponding to the brand to detect the camera model, and finally use the corresponding individual model to detect the source camera individual. Such experimental setup is acceptable because the difficulty levels of identifying camera brand, camera model and camera individual increase gradually. The hierarchical experimental setup contributes to improving the detection accuracy. The proposed framework is implemented in MATLAB 2013, and the experiments are carried out on the computer with Intel (R) Xeon (R) 2.6 GHz CPU, 128 GB RAM. All experiments in this paper are iterated 10 times to get the average accuracy.

3.1. Component analysis

In this section, component analysis experiments are performed to demonstrate the importance of each part of the proposed framework. To be specific, we remove the main modules of the proposed framework one by one for component analysis and explore various algorithm options in each module to find the combination that performs best. It is worth noting that when changing one module, the other modules of the framework remain fixed.

3.1.1. Image preprocessing

The experiments related to preprocessing are performed by using different preprocessing algorithms. For simplicity, in other modules of the framework, we use wavelet transform, LBP feature and LIBLINEAR classifier respectively, and set the feature dimension after PCA to 100. The detection accuracy for 20 cameras is shown in Table 2. It can be seen that “Demosaicing+Predicting+De-noising” achieves the highest accuracy when other modules of the framework are

Table 2
Detection accuracy using different preprocessing algorithms.

Preprocessing	Accuracy (%)
None	82.4
Demosaicing	83.1
Predicting	82.0
De-noising	83.1
Re-balancing	80.7
Demosaicing + Predicting	83.4
Demosaicing + De-noising	83.5
Demosaicing + Re-balancing	80.6
Predicting + De-noising	83.7
Predicting + Re-balancing	81.2
De-noising + Re-balancing	81.3
Demosaicing + Predicting + De-noising	**83.8**
Demosaicing + Predicting + Re-balancing	81.0
Predicting + De-noising + Re-balancing	81.6
Demosaicing + Predicting + De-noising + Re-balancing	81.7

The bold value indicates that the value is the highest.

fixed. Therefore, in the following experiments, we use demosaicing, predicting and de-noising as the preprocessing algorithms.

It is noteworthy that the execution of the three algorithms is parallel rather than serial, as shown in Fig. 9.

3.1.2. Residual image calculation

In this section, the importance of residual image calculation is demonstrated by removing this module. Similarly, we use wavelet transform, LBP feature and LIBLINEAR classifier in other modules, and set the feature dimension after PCA to 100. Demosaicing, predicting and de-noising are used in preprocessing. Experimentally, the detection accuracy for 20 cameras of the proposed framework without calculating the residual image is 78.7%, which is lower than the result of 83.8% in Table 2.

3.1.3. Image transform

Table 3 shows the detection accuracy for 20 cameras when different image transform algorithms are used in the proposed framework. The algorithms used by other modules is the same as above. From the comparison, we can see that the detection accuracies of using wavelet transform and contourlet transform are higher than that without image transform, which verifies that the effectiveness of image transform. Note that although the contourlet transform can obtain more directional subbands, its performance is found to be worse than the wavelet transform. The possible reason is that the contour transform may capture more scene information left in the residual images, which weakens the feature discrimination. Therefore, we can conclude that the image transform contributes to the improvement of detection accuracy,

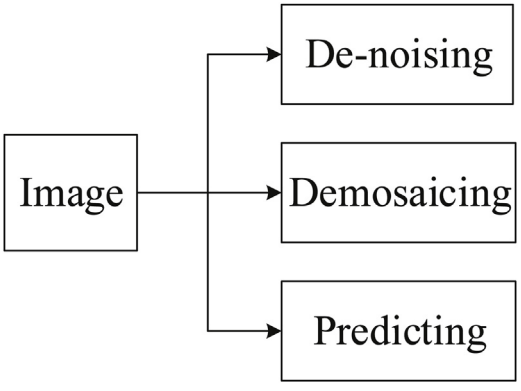

Fig. 9. The architecture of the preprocessing module.

Table 3
Detection accuracy using different image transform algorithms.

Image transform	Accuracy (%)
No transform	81.5
Wavelet	**83.8**
Contourlet	81.8
Wavelet + Contourlet	83.4

The bold value indicates that the value is the highest.

and image transform algorithms should be selected according to the model used.

3.1.4. Feature extraction

In Table 4, we show the detection accuracy for 20 cameras when different features are used. The combination of statistical moment features and LBP features gives us a satisfactory result. In the following sections, we use the combination of statistical moment features and LBP features for experiments.

3.1.5. Feature dimension reduction

The relationship between PCA feature dimension and accuracy is shown in Fig. 10. In general, if the PCA feature dimension $d \geq m - 1$, there is no information loss, where m is the number of training samples. Therefore, the PCA feature dimension used in this paper range from 0 to 2000. Specifically, we set the PCA feature dimension to 700, which perform best in the experiments.

3.1.6. Classification

So far, we have determined the algorithms used in the modules before classification, namely "Demosaicing+Predicting+De-noising", wavelet transform, "Statistical moment+LBP". The feature dimension after PCA is set to 700. In this section, some mainstream classifiers are used to classify the features obtained above. The results are shown in Table 5. Among the three classifiers used, Liblinear achieves the highest accuracy of 89

3.2. Comparison with the state-of-the-art

Based on the proposed framework and the experimental results in Section 3.1, we propose a new feature-based source camera identification method. The proposed method carries out demosaicing, predicting and de-noising algorithms in preprocessing module, performs a wavelet transform in image transform module, combines statistical moment features with LBP features in feature extraction module, uses Liblinear in classification module, and sets the feature dimension after PCA to 700. In order to prove the superiority of the proposed method, we compare our method with the state-of-the-art feature-based methods proposed in [21] and [23]. The experimental results of the proposed framework are shown in Fig. 11, which gives the confusion matrix of 20 cameras. The average detection accuracy of 20 cameras is 89.0%. The baselines are executed under the same experimental settings, and the comparison between the baselines and the proposed

Table 4
Detection accuracy using different features.

Feature	Accuracy (%)
IQM	77.6
Statistical moment	84.0
LBP	83.7
IQM + Statistical moment	84.2
IQM + LBP	83.3
Statistical moment + LBP	**85.0**
IQM + Statistical moment + LBP	84.8

The bold value indicates that the value is the highest.

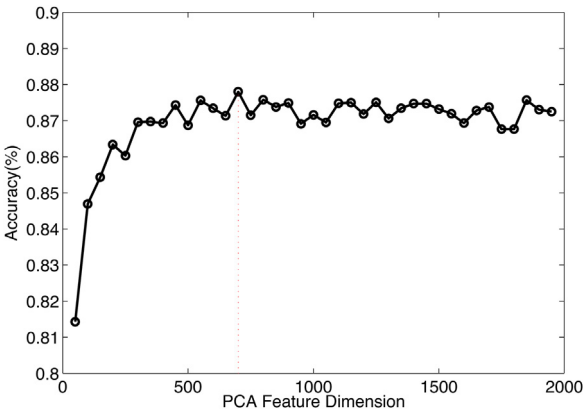

Fig. 10. Detection accuracy vs. PCA feature dimension plot. Accuracy is maximized when PCA feature dimension is 700.

Table 5
Detection accuracy using different classifiers.

classifier	Accuracy (%)
LIBSVM (RBF)	80.7
Liblinear	**89.0**
Random forest	77.6

The bold value indicates that the value is the highest.

framework is shown in Fig. 12. It can be seen that the proposed framework outperforms the method [21] and [23] for most of the cameras used. The average detection accuracies of the method [21] and [23] for 20 cameras are 75.7% and 84.1%, respectively, which are lower than that of the proposed framework.

Taking different camera models of the same camera brand in the confusion matrix as the same class, we can obtain the detection accuracy of proposed method for different camera brands, as shown in Table 6. It can be seen that all three methods can identify different camera brands with high accuracy, and the proposed method performs best. Similarly, when different camera individuals of the same camera model are taken as the same class, the detection accuracy of proposed method for 15 camera models are obtained, as shown in Table 7, which demonstrates the superiority of proposed method in identifying different camera models. Table 8 shows that the proposed method can also identify different camera individuals of the same model with higher accuracy.

We also record the training time and testing time of the classifiers for three methods when the above experiment is carried out once. The results are shown in Table 9. The classification time of the proposed framework is of the same order of magnitude as that of the method [21] and is much lower than the classification time of method [23]. Therefore, it can be concluded that the proposed framework can identify the source camera with higher accuracy and efficiency than the existing methods.

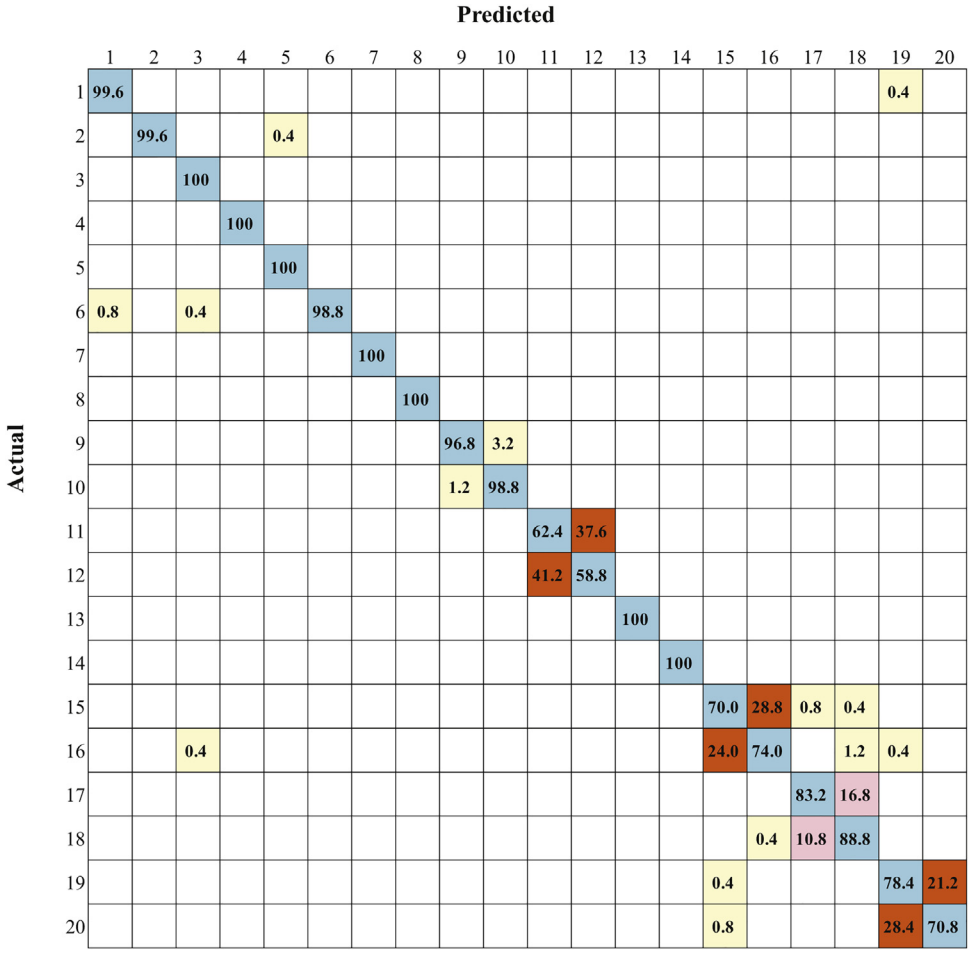

Fig. 11. Confusion matrix for 20 cameras of the proposed method.

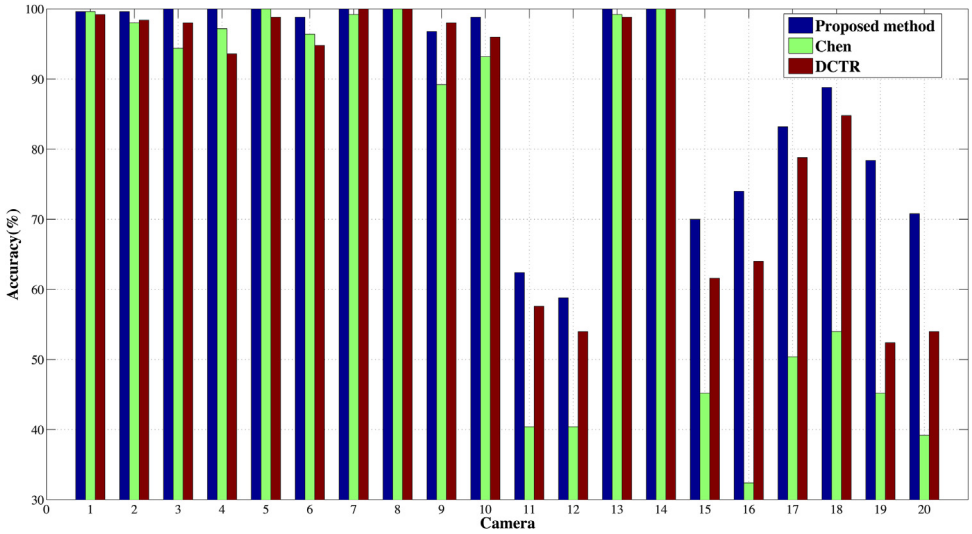

Fig. 12. Comparison of detection accuracy using three methods.

Table 6
Comparison of detection accuracy for camera brand (%).

Camera brand	Proposed	Chen [23]	DCTR [21]
Agfa	99.7	98.8	99.3
Canon	99.6	98.8	99.3
Casio	100	100	99.2
FujiFilm	100	100	100
Nikon	100	99.6	99.8
Olympus	100	98.8	99.2
Panasonic	100	100	100
Sony	99.9	99.7	99.2
Average	99.9	99.5	99.5

Table 7
Comparison of detection accuracy for camera model (%).

Camera model	Proposed	Chen [23]	DCTR [21]
Agfa_DC-504	99.6	99.2	99.6
Agfa_DC-733s	99.6	98.4	98
Agfa_DC-830i	100	98.0	94.4
Canon_Ixus55	100	93.6	97.2
Canon_Ixus70	100	98.8	100
Canon_PowerShotA640	98.8	94.8	96.4
Casio_EX-Z150	100	100	99.2
FujiFilm_FinePixJ50	100	100	100
Nikon_D200	100	99.4	99.6
Nikon_D70	100	97.6	99.6
Olympus_mju_1050SW	100	98.8	99.2
Panasonic_DMC-FZ50	100	100	100
Sony_DSC-H50	98.4	90.6	83.8
Sony_DSC-W170	99.8	98.6	80.4
Sony_DSC-W170	99.4	82.8	79.2
Average	99.7	96.7	95.1

Table 8
Comparison of detection accuracy for camera individual (%).

Camera individual	Proposed	Chen [23]	DCTR [21]
Nikon_D200	97.8	97.0	91.2
Nikon_D70	60.6	55.8	40.4
Sony_DSC-H50	72.0	62.8	38.8
Sony_ DSC-T77	86.0	81.8	52.2
Sony _DSC-W170	74.6	53.2	42.2
Average	78.2	70.1	53.0

Table 9
Comparison of classification time using three methods.

	Proposed	Chen [23]	DCTR [21]
Training (s)	259.3	5745.8	32.0
Testing (s)	0.9	22.2	0.1
Total (s)	260.2	5768	32.1

In our last experiment, we will demonstrate the effectiveness of the proposed hierarchical experimental setup. Just as most existing methods have done, we only train one classification model to identify the source cameras. The detection accuracy of the classification model for 20 cameras is 87.5%, which is lower than the result using the proposed hierarchical experimental setting.

4. Conclusion

In this paper, we propose a feature-based framework for source camera identification, which consists of six parts: image preprocessing, residual image calculation, image transform, feature extraction, feature dimension reduction, and classification. By experimenting with different algorithm options in the proposed framework, we propose a new source camera identification method that can more capture camera-related artifacts and represent them with more discriminative features. The proposed method has an average detection accuracy of 89% for 20 cameras, which is superior to the existing state-of-the-art methods. Besides, the importance of each module in the proposed framework has also been demonstrated by component analysis experiments. Our future work will focus on using new preprocessing algorithms or features to improve the performance of the proposed framework.

Authors' contributions

Bo Wang: Conceptualization and Methodology.
Kun Zhong: Software and Writing-Original draft preparation.
Zihao Shan: Writing-Reviewing and Editing.
MeiNeng Zhu: Supervision and Visualization.
XueSui: Validation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. U1936117 and No. 61772111).

References

[1] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 1 (2006) 205–214.
[2] M.K. Mihcak, I. Kozintsev, K. Ramchandran, Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising, Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3256 (1999) 3253–3256.
[3] M. Chen, J. Fridrich, M. Goijan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Trans. Inf. Forensics Secur. 3 (2008) 74–90.
[4] G. Wu, X. Kang, K.J.R. Liu, A context adaptive predictor of sensor pattern noise for camera source identification, Proc. of the IEEE International Conference on Image Processing (2012) 237–240.
[5] X. Kang, J. Chen, K. Lin, P. Anjie, A context-adaptive SPN predictor for trustworthy source camera identification, J. Image Video Proc. 2014 (2014) 1–11.
[6] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2012) 1397–1409.
[7] H. Zeng, X. Kang, Fast source camera identification using content adaptive guided image filter, Forensic Sci. 61 (2016) 520–526.
[8] C.-T. Li, Source camera identification using enhanced sensor pattern noise, Proc. of the IEEE International Conference on Image Processing (2009) 1493–1496.
[9] X.G. Kang, Y.X. Li, Z.H. Qu, J.W. Huang, Enhancing source camera identification performance with a camera reference phase sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 7 (2012) 393–402.
[10] X. Lin, C.-T. Li, Preprocessing reference sensor pattern noise via spectrum equalization, IEEE Trans. Inf. Forensics Secur. 11 (2016) 126–140.
[11] L.B. Zhang, F. Peng, M.M. Long, Source camera identification based on guided image estimation and block weighted average, Proc. of the International Workshop on Digital Watermarking (2016) 106–118.
[12] A. Lawgaly, F. Khelifi, Sensor pattern noise estimation based on improved locally adaptive DCT filtering and weighted averaging for source camera identification and verification, IEEE Trans. Inf. Forensics Secur. 12 (2017) 392–404.
[13] M. Kharrazi, H.T. Sencar, N. Memon, Blind source camera identification, Proc. of the IEEE International Conference on Image Processing, vols. 1–5 (2004) 709–712.
[14] T. Gloe, Feature-based forensic camera model identification, in: Y.Q. Shi (Ed.), LNCS 7228, Springer-Verlag, Berlin/Heidelberg, 2012, pp. 42–62.
[15] M.J. Tsai, C.S. Wang, J. Liu, J.S. Yin, Using decision fusion of feature selection in digital forensics for camera source model identification, Comput. Stand. Interfaces 34 (2012) 292–304.
[16] B. Wang, Y. Guo, X. Kong, F. Meng, Source camera identification forensics based on wavelet features, Proc. of the Intelligent Information Hiding and Multimedia Signal Processing (2009) 702–705.
[17] I. Celiktutan, M. Avcibas, Sankur, Blind identification of cellular phone cameras, Proc. SPIE, 6505 (2007) 65051H.
[18] G. Xu, Y.Q. Shi, Camera model identification using local binary patterns, Proc. of the IEEE International Conference on Multimedia and Expo (2012) 392–397.
[19] A.W.A. Wahab, P. Bateman, Conditional probability based camera identification, Int. J. Cryptol. Res. 2 (2010) 9.
[20] A.W.A. Wahab, A.T.S. Ho, S. Li, Inter-camera model image source identification with conditional probability features, Proc. of the 3rd Image Electronics and Visual Computing Workshop (IEVC2012), Kuching, Malaysia, 2012.
[21] A. Roy, R.S. Chakraborty, V.U. Sameer, R. Naskar, Camera source identification using discrete cosine transform residue features and ensemble classifier, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2017) 1848–1854.
[22] B. Wang, X. Kong, X. You, Source camera identification using support vector machines, in: G. Peterson (Ed.), Advances in Digital Forensics V, 2009, pp. 107–118.
[23] C. Chen, M.C. Stamm, Camera model identification framework using an ensemble of demosaicing features, 2015 IEEE International Workshop on Information Forensics and Security (WIFS) (2015) 1–6.
[24] Z.H. Deng, A. Gijsenij, J.Y. Zhang, Source camera identification using auto-white balance approximation, Proc. of the IEEE International Conference on Computer Vision (ICCV 2011) (2011) 57–64.
[25] K.R. Akshatha, A.K. Karunakar, H. Anitha, U. Raghavendra, D. Shetty, Digital camera identification using PRNU: a feature based approach, Digit. Invest. 19 (2016) 207–214.
[26] B. Xu, X. Wang, X. Zhou, J. Xi, S. Wang, Source camera identification from image texture features, Neurocomputing 207 (2016) 131–140.
[27] H. Gou, A. Swaminathan, M. Wu, Intrinsic sensor noise features for forensic analysis on scanners and scanned images, IEEE Trans. Inf. Forensic Secur. 4 (3) (2009) 476–491.
[28] T. Ojala, M. Pietikinen, T. Menp, Multiresolution grayscale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[29] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[30] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[31] T. Gloe, R. Bohme, The dresden image database for benchmarking digital image forensics, J. Digit. Forensic Pract. 3 (2010) 150–159.