On robustness of camera identification algorithms

Jarosław Bernacki¹

Received: 27 August 2019 / Revised: 8 May 2020 / Accepted: 27 May 2020 /

Published online: 5 September 2020

© The Author(s) 2020

Abstract

In this paper we consider the problem of a privacy threat enabling tracing digital cameras by the analysis of pictures they produced. As thousands of images are processed at a mass scale, the threat may apply to most users of digital cameras. We consider a state-of-the-art algorithm for digital camera identification proposed in Lucas et al. (IEEE Trans Inf Forensics Secur 1(2):205–214, 2006) and discuss strategies that can be used to bypass it, in order to make information about the camera unavailable. It turns out that many natural strategies like Gaussian blur, adding artificial noise or removing pixels' least significant bit from the image does not prevent the identification of a camera unless a huge loss of image details is suffered. On the other hand, we show a method to bypass the camera identification with a just marginally more complex, yet not intuitive, method namely cropping the image on the edges and resizing to the original size using Lanczos resampling.

Keywords Privacy · Hardwaremetry · Camera recognition · Camera fingerprint · Lanczos resampling

1 Introduction

Since digital cameras are easily accessible and relatively cheap, many people take dozens of pictures on a daily basis. Furthermore, the pictures are shared using social media services or photo portals with friends, but also with thousands of coincidental onlookers. Posting hundreds of photos made using the same digital camera or a smartphone allows us to be linked to a certain device or a model, hence has negative consequences from the privacy point of view. The more privacy-aware of the social media users remove the meta-data indicating the time, location and the device model using image editors, however the linking a picture with a device does not require the additional meta-data. This can expose users'

✉ Jarosław Bernacki
jaroslaw.bernacki@outlook.com

¹ Częstochowa University of Technology, al. Armii Krajowej 36, 42-200, Częstochowa, Poland

privacy to a serious threat if they post an image outside of their personalized social media profiles, identifying pictures' author may lead from slight inconvenience like personalized spam to serious criminal activity like stalking or user personalized phishing. Naturally, the techniques of linking a device to an user might serve as well crime prevention. One of the most popular issue in image processing and image forensics is recognizing the camera based on images and using it as a "digital fingerprint" or proof of presence. An algorithm proposed by Lukás et al.'s in [21] is considered to be most efficient and is commonly used by black- and whitehats alike. Many recent approaches using deep learning models use image denoising formula presented in this approach. The assumption of Lukás et al.'s algorithm is to calculate the so called Photo-Response Nonuniformity Noise (PRNU) for each photo. It is shown that PRNU is specific to each camera and serves as a fingerprint for each device. The variance in PRNU among the devices is caused by sensors' imperfections that occur during the manufacturing process and use over the lifetime. Lukás et al.'s algorithm is based on calculation of camera's fingerprint by denoising input image $\mathbf{I}$ with filter F and calculating noise residual $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$. The value of $\mathbf{N}$ is calculated for each camera separately and serves as unique camera's fingerprint. Such approach provides a good efficacy of camera recognition but is very time consuming, thus cannot be used at larger scales. The obtained fingerprint is a strong link between a set of pictures and a device, allowing identification of the camera. Moreover, it is not easily removable from the image without a serious loss of image quality. In this paper we examine the robustness of aforementioned algorithm to attempts of breaking the link between the image and the camera. We show that some of the most "intuitive" strategies, such as noising image, Gaussian blurring or removing pixels' least significant bit does not allow to bypass the Lukás et al.'s camera identification algorithm without degrading picture quality. On the other hand, we show a technique which allows to obscure the camera fingerprint, without significant computational overhead nor deterioration of the image. This work is a continuation of research described in [1] that was presented at SECURE 2017 conference.

1.1 Contribution

The contribution of this paper is twofold. First, we show that simple and natural strategies like adding salt and pepper noise, Gaussian blurring or removing least significant bit of pixel intensities do not prevent from camera identification in terms of Lukás et al.'s algorithm. Mentioned methods work only in the case of a significant deterioration of image quality. Second, we propose a method that is capable to bypass camera identification with negligible influence on image quality. Cropping and stretching image to the original quality can effectively "confuse" Lukás et al.'s algorithm. We show that it is sufficient to symmetrically remove just 6 pixels from each edge of the image and upsample it to the original size in order to deceive considered algorithm.

1.2 Organization of the paper

Everywhere in the paper a bold font denotes matrices or vectors. The paper is organized as follows. In Section 2 we refer to some of previous and related works. In Section 3 we give some preliminaries useful in the rest of the paper. In Section 4 we show that Lukás et al.'s algorithm is robust against some basic techniques. In Section 5 we present an adjustment of Lukás et al.'s algorithm for fragment of pictures. Section 6 describes a surprisingly simple protection method that prevents from camera identification. We conclude this work in Section 7.

2 Previous and related work

The issue of camera fingerprinting has been investigated in several aspects. Camera fingerprint can be removed in various ways. The study of PRNU robustness and reliability in matters of forensics is presented [30]. It is possible to counterfeit characteristic features of a camera in order to produce an image “pretending” to be done by another camera. Such approach was presented in [13], where a method for PRNU fingerprint-copy attack is described. The aim is to obfuscate the PRNU of a particular camera by “inserting” into it a PRNU of another camera. Authors show that this can be done by performing some simple algebraic operations. Assume that $\hat{\mathbf{K}}_N$ is a camera fingerprint calculated of N images and $\mathbf{J}$ is a fingerprint of another camera whose we want to put into $\hat{\mathbf{K}}_N$. Then we have $\mathbf{J}' = \mathbf{J}(1 + \alpha \hat{\mathbf{K}}_N)$, where $\alpha > 0$ is a scalar fingerprint strength. Extensive experimental verification shows that such “exchange” of camera fingerprints is efficient, i.e. based on $\hat{\mathbf{K}}_N$ we can produce a fake photo $\mathbf{J}'$ that pretends that camera. Quite similar approach is described in [38]. However, both approaches may not be practical, because it is needed to have a representative image set of camera that we want to “exchange” the fingerprint with and affects the actual image (i.e., the stored information). Moreover, it may be easily detected that there was an image forgery. Such approaches are also very time-consuming.

The PRNU Decompare [28] was a software proposed in 2014 that aimed to anonymize images by manipulating their PRNU. It removes, forges or even destroys the PRNU of the image without a negative impact on images’ quality. However, this software seems to be no longer supported and available.

In [17] a method using Color Filter Array (CFA) in order to create pattern in digital images is presented. Such solution can be used for concealing traces of image manipulation. As an image similarity measure, the Peak signal-to-noise (PSNR) ratio is used. However, proposed solution is appropriate only in case of CFA-based forensics methods. In [18] the camera fingerprint is described as an additive signal to images. It is proposed to remove the fingerprint with a fixed strange. However, better results can be achieved by using various strength, what was showed in [13]. An attempt of removing the sensor information was made by a method of flatfielding which is presented in [21]. In [1] a fast method for camera recognition as well as a method of preventing fingerprint linking in a grayscale images was proposed. The algorithm, based on Peak signal-to-noise ratio, is very fast in comparison with [21], however the classification efficiency is lower.

Recently, more and more approaches begin to use deep learning or convolutional neural networks (CNNs) in order to camera identification. An example of an algorithm based on use of convolutional neural networks is presented in [2]. Proposed algorithm learns features characterizing each camera from acquired images by performing operations through four convolutional layers. Results showed that such strategy outperforms former machine learning-based algorithms on classification of 64×64 pixels image patches. In [37] a robust multi-classifier based on CNN was proposed. The main advantage of proposed method is that it is capable to perform identification of multiple camera models in one comparison. The core of the method is similar to [2] i.e. images are pre-processed for patch selection. General accuracy of classification is high and usually reaches about 98%, however for some cameras proposed method achieves very poor results (accuracy of classification at 41%). Moreover, the time performance of this method and also of [2] is not examined in comparison for non-CNNs based algorithms.

Source camera identification based on deep learning is also investigated in [8]. Convolutional layers composed by a set of high pass filters are applied to the input image in order to extract and learn characteristic features. Authors conducted experiments on a set of modern

smartphones which are iPhone 5, Samsung Galaxy S4 and Samsung Galaxy Tab II. Accuracy of classification is very high (not least than 98% for each model), however the image dataset based on only three mentioned devices is not representative.

In [24] a framework for iris and camera model identification based on deep learning is discussed. Proposed algorithm uses convolutional neural networks for (1) iris and (2) camera sensor recognition. Proposed solution can be especially useful for the task of double-check user authentication, for instance for iris biometric logging into the authentication system with use of mobile devices.

Another approach using convolutional neural networks is presented in [34]. The network consists of three convolutional layers. The first layer is responsible for calculation of the noise pattern $\mathbf{N}$ which is extracted from images from the well-known formula $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$ [13, 16, 21]. $\mathbf{I}$ denotes the input image and F is the denoising filter. Noise patterns are treated with 64 kernels of size 3×3 and the size of produced feature maps is 126×126 . Second layer produces feature maps of size 64×64 . Third layer applies convolutions with 32 kernels of size 3×3 . The Rectified Linear Unit (ReLU) is applied to the output of every convolutional layer as an activation function. However, learning image features from denoising was proposed by Lukás et al. in 2006 which in this paper we consider as state-of-the-art [21]. Time for learning the network and image classification is not examined.

In [33] an analysis of image features effect on PRNU is described. In order to increase the quality of PRNU, a weighting function is proposed. Image regions which give reliable and unreliable PRNU are estimated and the weighting function gives weights for pixel-based correlation. Higher weights are assigned to image regions which provide reliable PRNU; smaller weights are assigned to regions giving less reliable PRNU. Experiments showed that proposed technique correctly described image features, what as a result slightly improved the accuracy of camera identification compared to [21].

In [15] a study on improving camera identification using PRNU is considered. It is proposed to extract PRNU from image separately, applied on low and high frequency components of the image. Results from the Dresden Image Database [12] showed that PRNU obtained with proposed method contain less amount of high frequency details of the images. However, it is not clearly shown, how such change improves the camera identification.

In [19] a compact representation of sensor fingerprint is introduced in order to reduce fingerprint processing computational costs. Proposed framework compresses the sensor pattern noises what significantly speeds up their processing time. The compact SPN representation can be obtained directly from a high-dimensional SPN. Experiments are held on Dresden Image Database [12] and reveal that compact representation of the fingerprint does not decrease the identification accuracy.

In [25] the mismatches of correlation values of multiple PRNUs are analyzed. Author provides a very simple formula that predicts the standard deviation of the mismatch of correlation values if they are close to 0 but do not belong to the particular camera. Proposed method is expected to achieve reliable results in camera identification, however it is not clearly described, how proposed method could be used to improve the camera identification accuracy.

In [36] a content-adaptive fusion residual networks is described. Images are divided into three categories: saturation, smoothness and others. For each category a fusion residual network is trained by transform learning. Experimental evaluation on Dresden Image Database [12] confirms high accuracy of proposed approach. In [29] the DenseNet convolutional network is proposed for camera identification. Deep learning for camera recognition is also used in [9], where a fully connected network is discussed.

Some denoising models introduce the so called attention mechanisms, therefore may be related with camera recognition. An attention model outputs the “summary” of the input vector or matrix. Attention models are also used for detecting salient objects in the images. Examples can be found in [6, 10, 20].

In [22] the vulnerability of deep learning approach to adversarial attacks for source camera identification is studied. It is considered how to produce fake photos in order to make a CNN-based classifier incorrectly classifying cameras. Several attacks are proposed where the image undergoes a lossless or a lossy compression. Sample attack scenarios include the following: the Fast Gradient Sign Method (FGSM) [14], DeepFool [26] and Jacobian-based Saliency Map Attack (JSMA) [27]. The idea of FGSM is quite simple and relies on adding an additive noise. However, in some case image quality can be visibly affected, therefore the DeepFool was introduced to overcome this problem. DeepFool is based on a local linearization of the classifier under attack. The JSMA is a greedy iterative procedure that computes a saliency map at each iteration and then marks the pixels that contribute most to the correct classification of the image. Experiments showed that applying such attacks efficiently deceive a CNN-based classifier.

3 Preliminaries

Below, we briefly recall the state-of-the-art camera identification algorithm we aim at protecting against and indicate most important literature referring to it. Then, we describe the image datasets that we have used in benchmarking our algorithms with.

3.1 Lukás et al.’s algorithm

Although the Lukás et al.’s algorithm was presented in 2006, to the best of our knowledge it is still the state-of-the-art method for camera identification and is widely considered as a reference point. There are a lot of research and patents that are based on this approach such as, among others [3, 16, 24, 38]. Also many recent approaches [23, 34] based on deep learning and convolutional neural networks use the denoising formula presented in this paper. The idea is to calculate for each camera the Photo-Response Nonuniformity Noise (PRNU), which is also called “noise residual” on a certain number of “training” images. Calculated noise residuals of images are averaged. The classification of a new photo is performed by calculating the **correlation value** between averaged noise residuals of a camera and noise residual of a new photo. Typically, the correlation value of image made by a particular camera is at least 0.01, on the other hand, if the image was not taken by a particular camera, the correlation value is usually much smaller, for example 0.002 or 0.0001. Therefore, the correlation already of ≥ 0.01 reveals that the picture matches the camera. This strategy is presented in Fig. 1. Formal description of Lukás et al.’s algorithm is presented in the [Appendix](#).

3.2 Image datasets

We have conducted experiments on two datasets of images.

Dataset I First dataset consist of modern devices such as smartphones, drones, compact cameras or DSLRs. The following devices were used: smartphones (Acer Liquid Jade S, Apple iPhone 5S, LG K10, Samsung Galaxy S7 and Samsung Galaxy Tab A (2016) tablet);

Fig. 1 Scheme of denoising images and classification of a new image [31]

compact cameras: Canon SX160, Canon SX270, Nikon P100); drones (DJI Spark, Yuneec Breeze 4K) and two DSLRs (Nikon D3100 and D7200). All devices contain CMOS imaging sensor except Canon SX160 (CCD). Image resolutions are from 8 to 14 megapixels for nearly all devices and 24 megapixels for Nikon D7200. There were 45 images per each device used.

Dataset II – The Dresden Image Database For a high-scale experiments, we have also used a large, public set of JPG images from the Dresden Image Database [12]. This database consists of tens of thousands images made by dozens of cameras and is often used for research or as a benchmark [5, 11]. We have used 11787 images from 48 cameras. The used cameras include: Agfa DC 733s, Agfa DC 830i, Agfa Sensor 505, Agfa Sensor 530s, Canon Ixus 55, Canon Ixus 70 (3 devices), Casio EX Z150 (5 devices), Kodak M1063 (5 devices), Nikon CoolPix S710 (5 devices), Nikon D70 (2 devices), Nikon D70s (2 devices), Nikon D200 (2 devices), Olympus 1050SW (5 devices), Praktica DCZ5 (5 devices), Rollei RCP 7325XS (3 devices), Samsung L74 (3 devices) and Samsung NV15 (3 devices). In the paper we aim to provide a method preserving users' privacy, hence we treat each instance of a specific camera separately (for example we distinguish between Canon Ixus 70 (1) and Canon Ixus 70 (2)).

4 Robustness of Lukás et al.'s algorithm against typical method of removing characteristic features

In this section we examine the robustness of the PRNU-based method against some basic image processing operations. We will check if the correlation value (introduced in Preliminaries section) between the calculated camera's noise residual (which in further part of this paper we will call **fingerprint**) and images after some transformations remain in the pattern indicating the photo's original device.

It turns out that many methods that intuitively should remove characteristic features of the camera does not work without re-iterating the processing (or increasing the parameter values) to the point of significant degradation of the quality of processed image. Therefore, the image may be useless for a wide range of applications, and Lukás et al.'s algorithm still allows for a correct linking between the image and the camera. In particular, we consider adding random noise to the images (also known as salt and pepper noise), using Gaussian blur and also removing pixels' least significant bit (LSB), i.e. setting the least significant bit of each of the pixels' intensity for each of the color channel to 0, independently of the previous value. The results show that even though the methods are applied until the quality of the image is strongly affected, the Lukás et al.'s algorithm still correctly identifies the camera.

4.1 Adding salt and pepper noise

We consider images represented in the RGB model, where pixel values for each color channel $\mathcal{R}$ (red), $\mathcal{G}$ (green) and $\mathcal{B}$ (blue) take values from $[0, 255]$. We propose to replace k pixels in the image, in a manner that $k/2$ pixels in the picture will be set to 0 and $k/2$ pixels to 255 in a random way. We examine the robustness of Lukás et al.'s algorithm against the procedure, measured as a correlation value peak for a correct camera's fingerprint, i.e., possibility of correct linking the noised image with the camera. We used MATLAB salt & pepper noise, implemented in `imnoise` function with the noise parameter k (which is probability of noise density) ranging from 0.001 to 1.0. Experiments were conducted for Dataset I and Dresden Image Database, referred to in Section 1. Mean correlation values calculated of all cameras can be seen in Fig. 2; also in Tables 1 and 2.

The results indicate that the fingerprint obfuscation is achieved at least at probability $k = 0.2$ (Dresden Image Database) and $k = 0.5$ (Dataset I). The correlation values are equal 0, but the image quality is strongly affected as depicted in Fig. 3.

Results indicate that image quality has to be strongly affected to achieve correlation value at 0.00. The probability of occurrence of random points in the image has to be not smaller than 0.2. Therefore, one may not assume the usability of such approach.

Fig. 2 Mean correlation values (Y-axis) according to increase of the salt and pepper density (X-axis)

Table 1 Correlation values for next densities of salt and pepper noise, Dataset I

Salt & pepper density	0.001	0.002	0.005	0.01	0.02	0.05	0.1	0.2 – 1.0
Acer Liquid Jade S	0.08	0.06	0.03	0.02	0.01	0.01	0.00	0.00
Canon SX160	0.07	0.07	0.06	0.06	0.05	0.05	0.04	0.00
Canon SX270	0.05	0.05	0.05	0.04	0.01	0.01	0.00	0.00
DJI Spark	0.09	0.08	0.08	0.08	0.06	0.05	0.03	0.00
iPhone 5S	0.07	0.06	0.07	0.06	0.06	0.04	0.03	0.00
LG K10	0.08	0.07	0.06	0.04	0.02	0.02	0.00	0.00
Nikon D3100	0.08	0.07	0.04	0.03	0.02	0.00	0.00	0.00
Nikon D7200	0.06	0.05	0.04	0.03	0.02	0.01	0.00	0.00
Nikon P100	0.03	0.03	0.02	0.02	0.01	0.01	0.01	0.00
Samsung Galaxy Tab	0.09	0.07	0.06	0.05	0.03	0.02	0.02	0.00
Samsung Galaxy S7	0.04	0.03	0.03	0.02	0.01	0.01	0.01	0.00
Yuneec Breeze 4K	0.07	0.07	0.04	0.03	0.04	0.02	0.01	0.00

4.2 Gaussian blurring

Gaussian smoothing, often referred to as Gaussian blur, is a standard and commonly used filter that is based on the normal distribution function (also named Gaussian kernel) with variance σ and mean equal 0. The σ defines the strength of blurring [7]. In our experiments, the goal is to convolve the Gaussian kernel with the input image $\mathbf{I}$ in order to calculate the output image $\mathbf{I}'$. Filter decreases the contrast between pixels, therefore it makes colors in image more “soft”.

Table 2 Correlation values for next densities of salt and pepper noise, Dresden Image Database

Salt & pepper density	0.001	0.002	0.005	0.01	0.02	0.05	0.1	0.2 – 1.0
Agfa DC 733s	0.16	0.15	0.13	0.10	0.06	0.13	0.01	0.00
Agfa DC 830i	0.06	0.06	0.05	0.04	0.02	0.01	0.00	0.00
Agfa Sensor 505	0.07	0.06	0.05	0.03	0.02	0.00	0.00	0.00
Agfa Sensor 530s	0.06	0.06	0.05	0.04	0.03	0.02	0.01	0.00
Canon Ixus 55	0.03	0.02	0.02	0.02	0.01	0.00	0.00	0.00
Canon Ixus 70	0.05	0.04	0.04	0.03	0.02	0.01	0.00	0.00
Casio EX Z150	0.02	0.02	0.02	0.01	0.01	0.00	0.00	0.00
Kodak M1063	0.06	0.05	0.04	0.03	0.02	0.01	0.00	0.00
Nikon CoolPix S710	0.08	0.07	0.06	0.04	0.03	0.02	0.00	0.00
Nikon D70	0.05	0.04	0.03	0.03	0.02	0.00	0.00	0.00
Nikon D70s	0.04	0.04	0.03	0.02	0.02	0.00	0.00	0.00
Nikon D200	0.04	0.04	0.03	0.03	0.02	0.01	0.00	0.00
Olympus 1050SW	0.04	0.04	0.03	0.03	0.02	0.00	0.00	0.00
Praktica DCZ5	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Rollei RCP 7325XS	0.05	0.05	0.05	0.04	0.03	0.01	0.00	0.00
Samsung L74	0.02	0.01	0.01	0.01	0.00	0.00	0.00	0.00
Samsung NV15	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00

Fig. 3 On the left: the original image (top) and image with random points noise at probability 0.2 (bottom). Camera: Kodak M1063. On the right: original image (top) and image with Gaussian blur at $\sigma = 0.001$. Camera: Samsung NV15

The experiments for image blurring were conducted with σ parameter equal 0.001, 0.01, 0.1, 0.2, 0.3.

The output image quality is comparable with that of the original image. Obviously, increasing the value of σ makes image quality worse and surprisingly, in some cases it also increases the correlation values. Thus such approach can not be used for a large-scale processing images in order to remove the fingerprint. Results are presented in Tables 3 and 4. The best results were achieved for $\sigma = 0.001$, since the correlation values for nearly all cameras were equal to 0.00, however, in some cases (DJI Spark, Agfa DC 733s and Agfa DC 830i) the correlations were as much as 0.1. In Fig. 3 can be seen a result of Gaussian blurring the image with $\sigma = 0.001$.

To sum up, Gaussian blurring cannot be considered as a stable solution for breaking the link between image and camera. Although image quality is better than in case of adding salt and pepper noise, the degradation is still noticeable. The correlation values for most values of parameter of Gaussian blurring are not small enough that one may assume the unlinkability between camera and the image.

4.3 Removing least significant bit

Yet another strategy is to remove pixels' least significant bit, namely setting the least significant bit of each of the pixels' intensity for each of the color channel to 0. For each pixel a LSB is removed with probability p , equal 0.001, 0.125, 0.25, 0.5 and 1.0. However, the results indicate that removing LSB does not decrease the correlation values for both datasets (Fig. 4).

Table 3 Correlation values for Gaussian blurring, Dataset I

σ	0.001	0.01	0.1	0.2	0.3
Acer Liquid Jade S	0.00	0.00	0.01	0.01	0.00
Canon SX160	0.01	0.00	0.00	0.00	0.01
Canon SX270	0.00	0.00	0.00	0.01	0.00
DJI Spark	0.00	0.01	0.10	0.01	0.01
iPhone 5S	0.01	0.01	0.00	0.00	0.00
LG K10	0.00	0.01	0.00	0.01	0.00
Nikon D3100	0.01	0.01	0.01	0.01	0.01
Nikon D7200	0.00	0.01	0.01	0.00	0.01
Nikon P100	0.01	0.00	0.00	0.00	0.00
Samsung Galaxy Tab	0.00	0.01	0.00	0.01	0.00
Samsung Galaxy S7	0.00	0.01	0.00	0.00	0.01
Yuneec Breeze 4K	0.00	0.01	0.01	0.00	0.01

Increasing the number of pixels with removed LSBs does not significantly decrease the correlation values, as the correlations even at level of 0.01 can allow camera identification. Of course, the image quality is strongly degraded as the example in Fig. 5 depicts.

As a conclusion it can be stated that removing least significant bit is not a reasonable operation for breaking the link between image and the camera, as the correlation values still stays in almost all cases at high values. Moreover, the image quality must be strongly affected, what makes image useless.

Table 4 Correlation values for Gaussian blurring, Dresden Image Database

σ	0.001	0.01	0.1	0.2	0.3
Agfa DC 733s	0.01	0.01	0.10	0.01	0.01
Agfa DC 830i	0.01	0.00	0.10	0.01	0.01
Agfa Sensor 505	0.00	0.00	0.00	0.01	0.01
Agfa Sensor 530s	0.01	0.00	0.01	0.01	0.01
Canon Ixus 55	0.00	0.00	0.01	0.01	0.00
Canon Ixus 70	0.01	0.00	0.01	0.01	0.01
Casio EX Z150	0.00	0.00	0.01	0.01	0.00
Kodak M1063	0.00	0.00	0.00	0.00	0.01
Nikon CoolPix S710	0.00	0.00	0.00	0.00	0.00
Nikon D70	0.00	0.00	0.00	0.01	0.01
Nikon D70s	0.00	0.00	0.01	0.01	0.00
Nikon D200	0.00	0.00	0.00	0.00	0.01
Olympus 1050SW	0.00	0.00	0.00	0.00	0.01
Praktica DCZ5	0.00	0.00	0.00	0.00	0.01
Rollei RCP 7325XS	0.00	0.01	0.01	0.00	0.01
Samsung L74	0.00	0.00	0.00	0.00	0.01
Samsung NV15	0.00	0.00	0.01	0.01	0.00

Fig. 4 Mean correlation values (Y -axis) according to increase probability of pixels with removed LSB (X -axis)

4.4 Numerical analysis of the deformation

We have analyzed the similarities of the original image $\mathbf{I}$ and noised images $\mathbf{I}'$. The residuum $\mathbf{D} = |\mathbf{I} - \mathbf{I}'|$ is a matrix of absolute values of pixel differences between original and noised image. Such residua have been calculated both for adding salt and pepper noise and Gaussian blur. Obviously, the higher random noise points' density, the worse image quality. The minimal probability for fingerprint obfuscation is 0.5 (Dataset I) and 0.2 (Dresden Image Database) but the image quality is strongly degraded. Simultaneously, in such case in nearly all cameras the correlation value is about 0.00. In case of Gaussian blurring, for most cameras sufficient probability is 0.001, where the image quality is relatively good. Detailed results about image quality are presented in Figs. 6 and 7.

Fig. 5 On the left: the original image $\mathbf{I}$; on the right: the image $\mathbf{I}'$ with LSB removed from all pixels. The mean of $\mathbf{D} = |\mathbf{I} - \mathbf{I}'|$ equals 59.14, standard deviation of D is 31.82, and the median is 61. Camera: Rollei RCP 7325XS

Fig. 6 Average difference, standard deviation and medians of pixel intensities for each tested camera between images $D = |I - I'|$ for salt and pepper noise. I stands for original image, I' is noised image

It is clearly visible, and quite intuitive, that increasing the density of salt and pepper decreases the correlation values between image with the original fingerprint. However, it also confirms that simple image deterioration by adding noise cannot successfully change the fingerprint in the manner that prevents from linking the camera with an image in terms of Lukás et al.'s algorithm.

4.5 Summary

In this section we have examined simple techniques for image noising in order to break the link between image and the camera in terms of Lukás et al.'s algorithm. We have investigated the following methods: adding random points to the image (also known as salt and pepper noise), Gaussian blurring and removing least significant bit. The analysis indicated that none of these techniques provided reasonable results in fooling Lukás et al.'s algorithm.

Fig. 7 Average difference, standard deviation and medians of pixel intensities for each tested camera between images $D = |I - I'|$ in Gaussian blur. I stands for original image, I' is blurred image

The worst results were obtained with salt & pepper noise and removing least significant bit, where image has to be strongly degraded to be not recognized by analyzed algorithm. Quite better results are observed in case of Gaussian blurring, where image is relatively acceptable quality. However, this approach seems to be not stable, because for some values of the parameter of Gaussian blurring recognition of the camera still remains strong.

5 Adjustment of Lukás et al.'s algorithm for fragments of pictures

We have analyzed the impact of dividing the image into sections to the possibility of camera identification. We have stated a question if it is possible to adopt Lukás et al.'s algorithm if

Fig. 8 Sample division of the image into sections

we have only a part of the picture (e.g a rectangle-shaped fragment of size $N' \times M'$) for some $N' \leq N$ and $M' \leq M$. Let us divide the image for nine sections, as it is shown in Fig. 8. The experiments have been conducted as follows. The fingerprint of 100 images have been calculated only for section A1. Then, we took an image not belonging to this fingerprint and we have divided it into nine sections: A1, A2, ..., A9, and followed by calculating the correlation values between the fingerprint of section A1 and image all sections.

Results showed in Fig. 9 indicate that the correlation values between fingerprint and the same image section are high enough to assume that a section of the image belongs to the same camera. However, correlations of other sections in nearly all cases do not allow for reasonable fingerprint identification. Correlation values are in general at the level of 0.00, in some cases at most 0.01. Detailed results are presented in Tables 5 and 6.

Fig. 9 Correlation values for next image sections (mean of all cameras of all datasets)

Table 5 Correlation values between camera's fingerprint and image sections; Dataset I

Camera	A1	A2	A3	A4	A5	A6	A7	A8	A9
Acer Liquid Jade S	0.17	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.01
Canon SX160	0.03	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00
Canon SX270	0.07	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00
DJI Spark	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
iPhone 5S	0.18	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00
LG K10	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon D3100	0.15	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon D7200	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon P100	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01
Samsung Galaxy Tab	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Samsung Galaxy S7	0.05	0.03	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Yuneec Breeze 4K	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00

For a smaller picture this approach works in a completely analogical way, although the correlation values are slightly smaller. To conclude, camera may be recognized by image sections only by analysis of the corresponding sections of the image. The correlation values between different image sections are low enough to be considered as made with a particular camera.

Table 6 Correlation values between camera's fingerprint and image sections; Dresden Image Database

Camera	A1	A2	A3	A4	A5	A6	A7	A8	A9
Agfa DC 733s	0.18	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00
Agfa DC 830i	0.17	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.01
Agfa Sensor505	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01
Agfa Sensor530s	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Canon Ixus 55	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Canon Ixus 70 1	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
Casio EX Z150 0	0.07	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00
Kodak M1063 1	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon CoolPix S710 1	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon D70 0	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon D70s 0	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon D200 1	0.15	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Olympus 1050SW 0	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Praktica DCZ5 1	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Rollei RCP 7325XS 1	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Samsung L74 1	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Samsung NV15 2	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Fig. 10 Correlation values for cropping next pixels (mean of all cameras)

6 Efficient quality-preserving prevention against Lukás et al.'s algorithm

As shown in Section 4, Lukás et al.'s algorithm is robust against many basic image processing methods aiming to break the link between an image and the device of its origin, however it is possible to break that link without huge losing the quality of pictures.

We have analyzed the possibility of camera identification in case of cropping the photo and stretching it to the original size by the following operations.

Input: Image I in RGB of size $M \times N$;

Output: Cropped image I' in RGB of size $M \times N$.

1. Crop the image by a certain number of pixels;
2. Upsample cropped photo to the original size using the Lanczos resampling algorithm.

Note that the final size of the processed image **has to** be the same as the original image. The requirement is first and foremost important for the PRNU-based algorithm, as it works only for the images of the same size, however it is important as well due to more practical reason – namely the adversary may easily check that the resolution does not match any

Fig. 11 In the left: the original photo of resolution 2592×1944 px, in the middle cropped along dimensions by 20px (5 pixels from each edge), in the right: cropped and stretched to the original size. Camera: Agfa Sensor 505

commercially available camera, thus it was tampered with. As shown in [32], the method of resampling the image may affect the image significantly. The Lanczos kernel was chosen for its negligible to the observer deterioration and introducing some artifacts that may influence PRNU. The idea of Lanczos resampling method is recalled in the [Appendix](#).

Experiments We have symmetric removed from 1 to 20 pixels from each edge of the photo and upsampled it to the original size. Obviously, it is associated with a slight loss of information, however there is no significant quality degradation. By cropping more pixels, the correlation values are gradually decreasing, while some objects located near the borders might be removed from the photo. Experiments showed that cropping each edge of a photo

Fig. 12 Average difference, standard deviation and medians of pixel intensities for each tested camera between images $D = |I - I'|$ in cropping experiments. I stands for original image, I' is cropped image

by 6 pixels prevents from identifying the original camera's fingerprint as the correlation results of Lukás et al.'s algorithm is equal to 0.00. Mean values of correlation values from devices of all datasets are shown in Fig. 10.

Cropping and “stretching” the image to its original size is very simple and fast. Moreover, cropping a small number of pixels does not degrade the picture quality or disturbs the geometry of the photographed shapes, one can interpret the resulting image as a photo that is slightly zoomed in comparison to the original. An example of an image after the procedure is presented in Fig. 11. General comparison of images quality is shown in Fig. 12.

It is important to note, that a procedure consisting of multiple iterations of Lanczos resampling (upsampling and downsampling) alone, i.e., without the previous cropping, also decreases the correlation values between the camera and the image (while comparing with the respective original image), however it does not always result in correlation value equal to zero. This means that in such cases camera can be successfully identified. Precise results are presented in Table 7.

Summary Results show that procedure of cropping the image already of 6 pixels by each edge and upsampling it to the original resolution defeats camera recognition of Lukás et al.'s algorithm. Such strategy is fast and almost imperceptible due to slight loss of original image information. We have also shown that multiple use of Lanczos resampling algorithm (for upsampling and downsampling the image without cropping) prevents from camera recognition. In this case the image quality is also not affected, what makes image useful.

Table 7 Correlation values for images resampled using Lanczos kernel (without cropping); all datasets

Dataset I		Dresden Image Database	
Camera	Lanczos	Camera	Lanczos
Acer Liquid Jade S	0.00	Agfa DC 733s	0.00
Canon SX160	0.01	Agfa DC 830i	0.03
Canon SX270	0.00	Agfa Sensor 505	0.00
DJI Spark	0.00	Agfa Sensor 530s	0.00
iPhone 5S	0.00	Canon Ixus 55	0.01
LG K10	0.01	Canon Ixus 70	0.00
Nikon D3100	0.00	Casio EX Z150	0.09
Nikon D7200	0.00	Kodak M1063	0.01
Nikon P100	0.05	Nikon CoolPix S710	0.01
Samsung Galaxy Tab	0.00	Nikon D70	0.02
Samsung Galaxy S7	0.00	Nikon D70s	0.00
Yuneec Breeze 4K	0.00	Nikon D200	0.01
		Olympus 1050SW	0.03
		Praktica DCZ5	0.00
		Rollei RCP 7325XS	0.00
		Samsung L74	0.00
		Samsung NV15	0.01

7 Conclusion

In this paper we focus on the topic of preserving the privacy of digital camera users, namely linking the photos with the device of its origin. We have shown that some of the typical methods are not sufficient for breaking the link between a photo and camera's fingerprint calculated as in widely used, state-of-the-art Lukás et al.'s algorithm. In order to achieve small correlation values between the image and the camera using those methods, it is necessary to significantly reduce the image quality. However, we have proposed a method which can successfully obscure the camera fingerprint. The method based on image cropping and resampling is showed to be efficient in breaking the link between the image and the device. Moreover, we have also analyzed the division of the image into sections to the possibility of camera identification.

As a future work, we plan to extend the experiments of camera identification by image sections. Such strategy is feasible as long as we know that the particular section of the image corresponds to the fingerprint of such section. The open problem is to identify the camera without particular knowledge of "location" of the fingerprint in the image. Another aspect to investigate concerns robustness of classification methods based on deep learning and convolutional neural networks. In particular, we are going to perform analogous experiments in order to check whether it is possible to "confuse" classification of CNNs-based approach by proposed methods.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix

Lukás et al.'s algorithm

The Lukás et al.'s algorithm is defined as follows.

Input: Image I in RGB of size $M \times N$;

Output: Matrix r of noise residual (fingerprint PRNU), size $M \times N$.

1. Denoise image I with denoising filter F ;
2. Calculate the matrix of noise residual: $r = I - F(I)$.

Matrix of noise residual r serves as the camera's fingerprint.

Lanczos resampling

Image resampling is often used as a part of preprocessing in order to adjust the height to width ratio or simply the size of the image. Utilizing a different resampling algorithm may result in significantly different outcomes as pointed out in [32]. Both up- and downsampling of an image results in reduction of the original information (e.g. averaging intensity values) as well as introduction of some "new information" specific to the used method.

Produced artifacts may not necessarily be a bad phenomenon as ringing artifacts occurring near significant intensity changes using Lanczos resampling [35] result in enhancement in the perceived image sharpness, despite adding noise that may obstruct a proper intensities or gradient analysis. In the following paper we utilize artifacts occurrence during image resampling in order to preserve privacy and hide the camera-specific features. Since Lanczos resampling not only introduces some noise, but also introduces ringing artifacts that increase the apparent sharpness and reduce the aliasing [35], hence resulting in negligible to the observer disturbance to the original image, we base our method on this particular resampling.

Lanczos resampling is an approximation (on a finite support) of a sinc filter. Namely, the resampling kernel that is convoluted with the image intensity values is a $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ function multiplied by a Lanczos window of width a . For a positive a , the kernel $L(x)$ has $2a - 1$ local extrema, a global one for $x = 0$ and a symmetric alternating local extrema responsible for the occurrence of the ringing artifacts. Below we remind the formula

$$\text{describing the kernel } L(x) = \begin{cases} 1 & \text{if } x = 0 \\ \frac{a \sin \pi x \sin \frac{\pi x}{a}}{\pi^2 x^2} & \text{if } -a \leq x < a \\ 0 & \text{otherwise} \end{cases}$$

When resampling, the signal samples are interpolated by a convolution with the kernel, that results in $S(x) = \sum_{i=\lfloor x \rfloor - a + 1}^{\lfloor x \rfloor + a} s_i L(x - i)$ [4]. Due to a values of 2 or 3 are most commonly used in both signal processing and commercial image editors, and $a = 2$ results in lowest perceived visual deterioration of the original image, we decided to use $a = 2$. Note that due to kernel maximum being equal to 1, the value of pixel intensity may never increase above 255 for a 8-bit channel, however for $|x| = 1.29671$ there are local minima at approximately -0.086 which may result up to negative intensity of 22, hence the final value of $S(x)$ ought to be normalized to $[0, 255]$ range.

References

1. Bernacki J, Klonowski M, Syga P (2017) Some remarks about tracing digital cameras – faster method and usable countermeasure. In: Proceedings of the 14th international joint conference on e-business and telecommunications (ICETE 2017), pp 343–350
2. Bondi L, Baroffio L, Guera D, Bestagini P, Delp EJ, Tubaro S (2017) First steps toward camera model identification with convolutional neural networks. *IEEE Signal Process Lett* 24(3):259–263
3. Bruno A, Cattaneo G, Ferraro Petrillo U, Narducci F, Roscigno G (2017) Distributed anti-plagiarism checker for biomedical images based on sensor noise. In: Battiatto S, Farinella GM, Leo M, Gallo G (eds) New trends in image analysis and processing – ICIAP 2017. Springer International Publishing, Cham, pp 343–352
4. Burger W, Burge MJ (2009) Principles of digital image processing core algorithms, 1st edn. Springer Publishing Company, Incorporated, New York
5. Deng Z, Gijssenij A, Zhang J (2011) Source camera identification using auto-white balance approximation. In: 2011 IEEE international conference on computer vision (ICCV), pp 57–64
6. Fan D, Liu J, Gao S, Hou Q, Borji A, Cheng M (2018) Salient objects in clutter bringing salient object detection to the foreground. *CoRR*, abs/1803.06091
7. Flusser J, Farokhi S, IV CH, Suk T, Zitová B., Pedone M (2016) Recognition of images degraded by gaussian blur. *IEEE Trans Image Process* 25(2):790–806
8. Freire-Obregón D, Narducci F, Barra S, Santana MC (2017) Deep learning for source camera identification on mobile devices. *CoRR*, abs/1710.01257
9. Freire-Obregón D, Narducci F, Barra S, Santana MC (2019) Deep learning for source camera identification on mobile devices. *Pattern Recogn Lett* 126:86–91
10. Fu K, Zhao Q, Gu IY, Yang J (2019) Deepside: a general deep framework for salient object detection. *Neurocomputing* 356:69–82

11. Gloe T (2012) Feature-based forensic camera model identification. LNCS Transactions on Data Hiding and Multimedia Security (DHMMS), to appear
12. Gloe T, Böhme R (2010) The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 25th symposium on applied computing (ACM SAC 2010), vol 2, pp 1585–1591
13. Goljan M, Fridrich JJ, Chen M (2011) Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Trans Inf Foren Sec* 6(1):227–236
14. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: 3rd international conference on learning representations, ICLR 2015, San diego, CA, USA, May 7-9, 2015, Conference track proceedings
15. Gupta B, Tiwari M (2018) Improving source camera identification performance using dct based image frequency components dependent sensor pattern noise extraction method. *Digit Invest* 24:121–127
16. Kang X, Li Y, Qu Z, Huang J (2012) Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Trans Inf Foren Sec* 7(2):393–402
17. Kirchner M, Böhme R (2009) Synthesis of color filter array pattern in digital images. In: Media forensics and security I, part of the IS&T-SPIE electronic imaging symposium, San Jose, CA, USA, January 19-21, 2009, Proceedings, 72540K
18. Li C, Chang C, Li Y (2009) On the repudiability of device identification and image integrity verification using sensor pattern noise. In: Information security and digital forensics - first international conference, ISDF 2009, London, United Kingdom, September 7-9, 2009, Revised selected papers, 19–25
19. Li R, Li C, Guan Y (2018) Inference of a compact representation of sensor fingerprint for source camera identification. *Pattern Recogn* 74:556–567
20. Liu J, Hou Q, Cheng M (2020) Dynamic feature integration for simultaneous detection of salient object, edge and skeleton. *CoRR*, abs/2004.08595
21. Lukás J, Fridrich JJ, Goljan M (2006) Digital camera identification from sensor pattern noise. *IEEE Trans Inf Foren Sec* 1(2):205–214
22. Marra F, Gragnaniello D, Verdoliva L (2018) On the vulnerability of deep learning to adversarial attacks for camera model identification. *Sig Proc Image Comm* 65:240–248
23. Marra F, Poggi G, Sansone C, Verdoliva L (2015) Evaluation of residual-based local features for camera model identification. In: New trends in image analysis and processing - ICIAP, 2015 workshops - ICIAP 2015 international workshops: BioFor, CTMR, RHEUMA, ISCA, MADiMa, SBMI, and QoEM, Genoa, Italy, September 7-8, 2015, Proceedings pp 11–18
24. Marra F, Poggi G, Sansone C, Verdoliva L (2017) A deep learning approach for iris sensor model identification. *Pattern Recognit Lett* 113:46–53
25. Mieremet A (2019) Camera-identification and common-source identification: The correlation values of mismatches. *Forensic Sci Int* 301:46–54
26. Moosavi-Dezfooli S, Fawzi A, Frossard P (2016) Deepfool: A simple and accurate method to fool deep neural networks. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp 2574–2582
27. Papernot N, McDaniel PD, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: IEEE european symposium on security and privacy, euroS&P saarbrücken, Germany, March 21-24, 2016, pp 372–387
28. PRNU Decompare (2014) Online; Accessed 18 March 2019
29. Rafi AM, Kamal U, Hoque R, Abrar A, Das S, Laganière R., Hasan MK (2019) Application of densenet in camera model identification and post-processing detection. In: IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2019, long beach, CA, USA, June 16-20, 2019, pp 19–28
30. Steinebach M, Liu H, Fan P, Katzenbeisser S (2010) Cell phone camera ballistics: attacks and countermeasures
31. Sutcu Y, Bayram S, Sencar HT, Memon ND (2007) Improvements on sensor noise based source camera identification. In: ICME. IEEE, Piscataway, pp 24–27
32. Syga P, Sieluzycy C, Krzyzanowska-berkowska P, Iskander DR (2017) Fully automated detection of lamina cribrosa in optical coherence tomography: Framework and illustrative examples. In: 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC), Jeju Island, South Korea, July 11-15, 2017, pp 608–611
33. Tiwari M, Gupta B (2018) Image features dependant correlation-weighting function for efficient prnu based source camera identification. *Forensic Sci Int* 285:111–120
34. Tuama A, Comby F, Chaumont M (2016) Camera model identification with the use of deep convolutional neural networks. 12
35. Turkowski K (1990) Graphics gems. chapter filters for common resampling tasks. Academic Press Professional, Inc, San Diego, pp 147–165

36. Yang P, Ni R, Zhao Y, Zhao W (2019) Source camera identification based on content-adaptive fusion residual networks. *Pattern Recogn Lett* 119:195–204
37. Yao H, Qiao T, Xu M, Zheng N (2018) Robust multi-classifier for camera model identification based on convolution neural network. *IEEE Access* 6:24973–24982
38. Zeng H, Chen J, Kang X, Zeng W (2015) Removing camera fingerprint to disguise photograph source. In: 2015 IEEE international conference on image processing, ICIP 2015, Quebec City, QC, Canada, September 27–30, 2015 pp 1687–1691

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