Multimedia Tools and Applications (2021) 80:29657–29673
https://doi.org/10.1007/s11042-021-11129-y

Robustness of digital camera identification with convolutional neural networks

Jarosław Bernacki[1]

Received: 26 October 2020 / Revised: 21 May 2021 / Accepted: 3 June 2021 /
Published online: 7 July 2021

Abstract
This paper considers the area of digital forensics (DF). One of the problem in DF is the issue of identification of digital cameras based on images. This aspect has been attractive in recent years due to popularity of social media platforms like Facebook, Twitter etc., where lots of photographs are shared. Although many algorithms and methods for digital camera identification have been proposed, there is lack of research about their robustness. Therefore, in this paper the robustness of digital camera identification with the use of convolutional neural network is discussed. It is assumed that images may be of poor quality, for example, degraded by Poisson noise, Gaussian blur, random noise or removing pixels' least significant bit. Experimental evaluation conducted on two large image datasets (including Dresden Image Database) confirms usefulness of proposed method, where noised images are recognized with almost the same high accuracy as normal images.

Keywords Digital forensics · Privacy · Hardwaremetry · Camera recognition · Camera fingerprint · Convolutional neural networks · Robustness

1 Introduction

Digital forensics is a popular area that attracts many scientific attention. Problems like identification of imaging sensors are especially interesting. One of the most challenging issue in digital forensics (and also in image processing) is identification of camera based on images and considering it as a "digital fingerprint" or a proof of presence. This domain is called by a term *hardwaremetry* [13]. Camera identification may be realized in two aspects. First is called the *individual source camera identification* (ISCI) which distinguishes a certain camera among cameras of both the same and the different camera models. The second aspect is called the *source camera model identification* (SCMI) that distinguishes a certain camera model among the different models but not distinguishes a certain copy of camera among other cameras of the same model. As example, for the following cameras: Sony A7 (0), Sony A7 (1), ... Sony A7 (n), Nikon D750 (0), Nikon D750 (1), ... , Nikon D750 (n),

✉ Jarosław Bernacki
jaroslaw.bernacki@pcz.pl

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

the ISCI distinguishes **all** cameras as different (Sony A7 (0), Sony A7 (1), ...), while the SCMI distinguishes only the general model (Sony A7, Nikon D750). Therefore, the ISCI is much stronger than SCMI aspect, therefore much research has been conducted in this domain [16–18, 22–24, 27]. One of the most popular algorithm for individual source camera identification is proposed by Lukás et al. [27]. This algorithm aims to identify cameras on so called Photo-Response Nonuniformity Noise (PRNU) which is also called sensor pattern noise or noise residual. The goal is to calculate $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$ where $\mathbf{N}$ is a noise residual, $\mathbf{I}$ is an input image and F is a denoising filter [21]. The $\mathbf{N}$ is unique for every camera and may be used for identification. Experimental evaluation confirmed very high camera identification accuracy. Recent years have shown another interest in field of camera identification thanks to convolutional neural networks (CNN) [4, 11, 31, 38, 40, 43]. Due to their nature, CNNs offer almost perfect classification accuracy in different subjects, as text or image classification and pattern recognition.

In this paper the robustness of digital camera identification with the use of a proposed convolutional neural network is discussed. The robustness is understood as a recognition of a camera based on visually affected images. More precisely, the network is learned by "normal" images of some camera and tested by applying to it images of the same camera degraded by Poisson noise, Gaussian blur, random noise and removing least significant bit (LSB) of pixel intensities. Results indicate that network successfully identifies even strongly affected images as coming from a particular camera. Discussed CNN may be also used for a digital camera identification. For evaluation, two large image datasets are used. First dataset includes modern cameras including latest digital single lens reflex/mirrorless, compact cameras and smartphones; second one is a Dresden Image Database [14] that is often used for benchmarking.

This paper is a continuation of research presented in [3] and in [2]. In [3] there have been proposed two algorithms called PSNR-CT and DEPECHE. The PSNR-CT algorithm is used for an ultra-fast camera identification (compared to Lukás et al. [27]). The DEPECHE algorithm is used for prevention of camera identification based on the analysis of image histogram. In [2] the robustness of camera identification in terms of Lukás et al. [27] by analysing degraded images has been checked. The degradation techniques included noising, blurring, removing least significant bit; also an algorithm to bypass the identification of Lukás et al.'s algorithm has been proposed. However, in [2] the impact of image degradation techniques to camera identification of other than Lukás et al.'s algorithm – for example convolutional neural networks was not examined. Therefore this is the motivation for this paper.

1.1 Contribution

The primary contribution of this paper is a study of robustness of digital camera identification with a convolutional neural network (CNN). This analysis covers an interference with the image quality by applying strategies like Poisson noise, Gaussian blur, random noise and removing pixels' least significant bit (LSB). It is showed that even strongly degraded images are still recognized by a CNN, therefore the identification of the original camera that produced the image is possible. Experiments are performed with the use of a proposed CNN that also might be used for a digital camera identification on not degraded images.

1.2 Organization of the paper

In next section the previous and related work are recalled. Section 3 depicts proposed convolutional neural network architecture. In Section 4 the experimental evaluation of proposed

method is described. Finally, the last section concludes this work. In the Appendix there is presented the code for implementation of proposed convolutional neural network under Python programming language. Everywhere in the paper, bold font denotes matrices or vectors.

2 Previous work

One of the most popular algorithm for digital camera fingerprinting was proposed by Lukás et al. [27]. This algorithm utilizes a Photo-Response Nonuniformity Noise (PRNU) which is unique for each camera and may serve as a fingerprint. The PRNU is also called as sensor pattern noise or noise residual. The PRNU is calculated as $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$ where $\mathbf{N}$ is a noise residual, $\mathbf{I}$ is an input image and F is a denoising filter. This method was further researched in [16–18, 22, 29]. In Marra et al. [29] image residuals were utilized for camera identification. The co-occurrence matrices of selected neighbors were used for features extraction. Residual images were calculated simply as the difference between the input image $\mathbf{I}$ and its denoised version $F(\mathbf{I})$; F was the denoising filter. Obtained features were applied as input of SVM classifier.

In Morshedi et al. [20] an algorithm for recognizing camera's sensor from High Dynamic Range (HDR) images was described. It was proposed to invert geometric transformations for enabling proper PRNU detection. Considered were reversal of upsampling and the patchwork. Experiments held on the UNIFI dataset [35] of HDR images from number of modern smartphones confirmed high accuracy of camera identification which was at least of 95%.

Agarwal et al. [1] described an algorithm for iris sensor identification. Image feature selections were collected by a Block Image Statistical Measure (BISM), High Order Wavelet Entropy (HOWE), Texture Measure (TM), Single-level Multi-orientation Wavelet Texture (SlMoWT) and image quality measures.

In Li et al.'s [26] investigation of camera identification by compact representation of fingerprints was discussed. Proposed algorithm generates compact representation of camera's fingerprints with the use of random projections strategy. Experiments showed that such approach may be practical and efficient, however robustness of proposed method was not checked.

Goljan et al. [15] discussed the effect of compression on camera identification using sensor fingerprint. Results indicated that the JPEG compression both increased the variance of the normalized correlation and the variance of peak-to-correlation energy (PCE).

Taspinar et al. [36] considered seam-carved images. Seam-carving is understood as removing some parts of the image. Results pointed that sensor recognition could be realized if the image block was even less than 50×50px. Another strategy includes the analysis of the generalized noise in natural images [37]. Proposed model utilizes parameters of the image that may be used as camera fingerprint. Tuama et al. [39] proposed training a machine learning classifier on the concatenation of the co-occurrences of color band noise residuals with features computed with a Markovian model in discrete cosine transform (DCT) domain. These features include also conditional probability statistics. Such model gives high order statistics which supplement and enhance the identification rate.

Vulnerability of deep learning approach to adversarial attacks was examined in Marra et al. [28]. It was discussed whether it is possible to deceive a CNN-based classifier in order to make camera classification incorrect. Attacking a CNN-based classifier was performed by the following methods: The Fast Gradient Sign Method (FGSM) [19], DeepFool [32] and Jacobian-based Saliency Map Attack (JSMA) [33]. The goal of FGSM was very simple and

relied on adding to images an additive noise. DeepFool is relied on a local linearization of the classifier. The JSMA was a greedy iterative procedure that detecting and replacing the pixels that contribute most to the correct classification of the image.

Obregon et al. [12] presented a fully connected network. Feature maps were obtained as a convolution of image **I** and kernel **K**. The network utilized two convolutional layers, a max pooling layer and three fully connected layers with ReLU function used for activation. Experiments held on MICHE dataset [7] with used hardware nVidia Tesla K80 (24GB) confirmed high accuracy of classification. Convolutional Neural Networks were also discussed in [5, 8, 25, 30, 34, 41]. In Yang et al. [42] a concept of using content-adaptive fusion residual networks was proposed. Images were divided into three categories: saturation, smoothness and others. For each image category a fusion residual network was trained by transform learning approach. Chen et al. [6] proposed a residual neural network (ResNet). The properties of camera's lens system were used for training the network. Discussed method was evaluated of individual source camera's identification in Dresden Image Database [14]. However, none of listed papers investigate the aspect of robustness of digital camera identification. Thus if the image was degraded by for example an adversarial attack in order to fool the classifier, the response of the classifier is not known.

3 Proposed CNN for individual source camera identification

3.1 Convolutional Neural Networks (CNN) – the background

Convolutional neural networks (CNN) are recently very popular in many fields. They are used for natural language processing, object/pattern recognition, different classification tasks including text or image classification. The general structure of a convolutional neural network includes layers containing the neurons. A neuron simply takes some value as input, does computations and returns the results to the next layer. Let us shortly recall the idea of CNNs. In contrary to traditional multilayer perceptron architecture, it uses two operations called *convolution* and *pooling* to reduce an image into essential features for further understanding and classifying the image. The general blocks of CNNs are convolution, activation, pooling and fully connected layers. The convolution layer (also named a filter) is passed over the image, viewing a few pixels at a time (for instance, 3×3 or 5×5). The convolution operation is a dot product of the input pixel values with weights defined in the filter. The results are summed up into one number that represents all the pixels observed by the filter. The result of convolution layer processing is passed to the activation layer. The activation layer takes as input the result of the convolution layer to find non-linearity in order to train the network itself using backpropagation. The most common activation function is Rectified Linear Units (ReLU) function, defined as $f(x) = \max(0, x)$. The activation function is applied to each value of the input image. The pooling layer stands for downsampling and reducing the size of the matrix. A filter is passed over the results of the previous layer and takes one number of each group, usually the maximum (often named a max-pooling layer), but in some cases the average. The goal of this operation is to focus on the most important information in each feature of the image, what allows to train the network much faster. Finally, the fully connected layers stand for a traditional multilayer perceptron architecture which input is a one dimensional vector representing the output of the previous layers. The output of the fully connected layer is a list of probabilities for different possible labels assigned to the image, usually calculated by the softmax function. The label with highest probability is the classification decision. The idea of CNN is presented in Fig. 1.

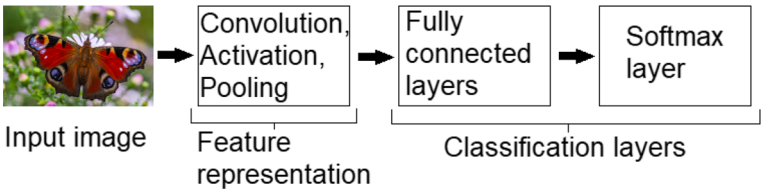

Fig. 1 The concept of Convolutional Neural Networks (CNN)

3.2 Proposed CNN

The digital camera identification can be realize with a following convolutional neural network. In contrary to [38], where the network is learned with $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$ ($\mathbf{N}$ is a noise residual, $\mathbf{I}$ is an input image and F is a denoising filter), proposed network may be learned directly with JPEG images without any additional procedures. The proposed network has three convolutional and two fully connected layers. As may be found in papers by Bondi et al.'s [4] and Yao et al.'s [43], we propose taking patches of size $64 \times 64 \times 3$ as input. The network structure is depicted below, full Keras implementation source code (under Python programming language) is presented in the Appendix.

1. First convolutional layer of 32 filters with kernel 5×5 and stride 1 with ReLU as an activation method;
2. A max-pooling layer with pool size of 2×2 and stride 2;
3. A second convolutional layer of 64 filters with kernel 5×5 with ReLU as an activation method;
4. A max-pooling layer with pool size 2×2;
5. A third convolutional layer of 128 filters with kernel 5×5 with ReLU as an activation method;
6. A max-pooling layer with pool size 2×2;
7. Two fully connection layers for classification: first fully connected layer with 4096 neurons with ReLU as an activation function and second fully connected layer with the output followed by the softmax function.

An input image $\mathbf{I}$ is passed to the first convolutional layer, consisting of 32 filters with kernel 5×5 with stride 1. Then, ReLU function is used as an activation method and a max-pooling layer with a pool size of 2×2 and stride 2 is applied. The second convolutional layer consists of 64 filters with kernel 5×5. Also the ReLU is used as an activation method and the max-pooling layer with pool size of 2×2. The third convolutional layer consist of 128 filters of kernel 5×5 with ReLU as activation function and max-pooling of size 2×2. Results are passed to the fully connected layers to obtain the final classification. First fully connected layer consists of 4096 neurons and ReLU is applied as an activation function to its output. Second fully connected layer activated with the softmax function provides the final classification.

4 Experimental evaluation

The proposed CNN has been evaluated in two experiments. Firstly (Experiment I), the robustness of the proposed CNN in terms of image degradation was examined. The network

was learned by normal images. For the classification, the Poisson noised, Gaussian blurred, random noised and least significant bit-removed images were applied to check whether the network will correctly identify the devices. Secondly (Experiment II), an experiment for typical individual source camera identification (ISCI) was conducted. Both experiments were performed on two image datasets which are described in next subsection. Images are represented in the RGB model, where pixel values for each color channel R (red), G (green) and B (blue) take values from [0, 255]. Scripts for image noising were implemented in Matlab, using the `imnoise` function.

As evaluation, the standard *accuracy* (ACC) and *true positive rate* (TPR) measures are used, defined as:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \quad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

where TP/TN denotes "true positive/true negative"; FP/FN stands for "false positive/false negative". TP denotes number of cases correctly classified to a specific class; TN are instances that are correctly rejected. FP denotes cases incorrectly classified to the specific class; FN are cases incorrectly rejected. As hardware, a notebook with Intel Core i5-7300HQ@2.5-3.1GHz CPU with 24 gigabytes of RAM (DDR4-2400) and nVidia GeForce GTX1050 GPU with 4 gigabytes of video memory has been used.

Experiments were held with 100 epochs for training and batch size of size 32. The number of training epochs and batch size was defined experimentally. Experiments showed that the number of 100 epochs is sufficient to successfully train the CNN and obtain the satisfactory classification accuracy. Due to large number of tested devices, the full results of camera model identification are not presented for clarity – instead of this there are presented confusion matrices only for brand recognition. In all tables serving as confusion matrices, rows denote the actual classes, columns denote the prediction results.

4.1 Image datasets

Dataset I First dataset contains images of modern devices such as smartphones, compact cameras or digital single lens reflex/mirrorless (DSLRs/DSLMs). This dataset includes the following models: Apple iPhone 8 (main and tele camera) (A1 and A2), Apple iPhone Xr (A3), Canon 1D X Mark II (C1), Canon 5D Mark IV (C2), Canon 90D (C3), Canon G9X Mark II (C4), Canon M6 Mark II (C5), Canon M10 (C6), Canon M100 (C7), Canon R (C8), Canon RP (C9), Huawei P9 Plus (H1), Huawei P20 Pro (H2), Huawei P20 Pro artificial intelligence-based camera (H3), Nikon D3X (N1), Nikon D5 (N2), Nikon D500 (N3), Nikon D610 (N4), Nikon D750 (N5), Nikon D810 (N6), Nikon D850 (N7), Nikon D7200 (N8), Nikon D7500 (N9), Nikon Z6 (N10), Nikon Z7 (N11), Panasonic Lumix GX800 (P1), Panasonic Lumix S1 (P2), Samsung S9 Plus (main – S1 and tele camera – S2), Samsung S10 Plus (main – S3, tele – S4 and ultrawide – S5 camera), Sony A7R III (S6), Sony A7S (S7), Sony A7S II (S8), Sony A9 (S9), Sony A6500 (S10), Sony RX100 II (S11), Sony Xperia 1 (main – S12 and ultrawide – S13 camera), Sony Xperia XZ1 (S14), Xiaomi Mi 9 (tele – X1, ultrawide – X2 and wide – X3) camera. At least 30 images per device were utilized, therefore total number of images is 1919 from 46 devices.

Dataset II – Dresden Image Database Also the Dresden Image Database [14] which is a popular set of JPG images has been used. This database consists of thousands of images made by plenty of cameras. There have been used 11787 images from 48 cameras, therefore at least 200 images from each device were used. The utilized cameras include: Agfa DC

733s (Ag1), Agfa DC 830i (Ag2), Agfa Sensor 505 (Ag3), Agfa Sensor 530s (Ag4), Canon Ixus 55 (Ca1), Canon Ixus 70 (3 devices - Ca2, Ca3, Ca4), Casio EX Z150 (5 devices - Ca5, Ca6, Ca7, Ca8, Ca9), Kodak M1063 (5 devices - Ko1, Ko2, Ko3, Ko4, Ko5), Nikon CoolPix S710 (5 devices - Ni1, Ni2, Ni3, Ni4, Ni5), Nikon D70 (2 devices - Ni6, Ni7), Nikon D70s (2 devices - Ni8, Ni9), Nikon D200 (2 devices - Ni10, Ni11), Olympus 1050SW (5 devices - Ol1, Ol2, Ol3, Ol4, Ol5), Praktica DCZ5 (5 devices - Pr1, Pr2, Pr3, Pr4, Pr5), Rollei RCP 7325XS (3 devices - Ro1, Ro2, Ro3), Samsung L74 (3 devices - Sa1, Sa2, Sa3) and Samsung NV15 (3 devices - Sa4, Sa5, Sa6).

As mentioned in previous sections, proposed CNN is tested in the aspect of individual source camera identification (ISCI), therefore the different copies of the same camera model, for example Nikon D200 (Ni10) and Nikon D200 (Ni11), etc are distinguished. The CNN was trained twofold, for each dataset independently. The relation between the number of images used for training and testing is the following: 80% of images used for training and 20% of images for testing.

4.2 Experiment I – Robustness of proposed CNN to image degradation operations

The analysis of robustness of proposed CNN to image degradation operations such as Poisson noising, Gaussian blurring, adding random noise and removing pixels' least significant bit (LSB) was conducted. In this experiment, the network was learned by normal images and for evaluation, was given images degraded with aforementioned methods. Some examples of image degradation can be seen in Figs. 2, 3, 4 and 5.

Poisson noise Poisson noise (also called quantum noise) is a signal-dependent noise that can be seen on images. Pixels x are generated discretely according to the Poisson distribution P (1):

$$P(x) = \frac{e^{-k}k^x}{x!} \tag{1}$$

where k is the mean parameter which in case of RGB images takes the same values as processed pixel [9]. A sample Poisson-blurred image can be seen as Fig. 2. Results of identification for Poisson noised images are presented in Tables 1 and 2.

Analysis of classification confirms that Poisson noising cannot be considered as a strategy for ensuring the unlinkability between the camera and the image. The accuracy of 99% for both datasets is the same as identification of normal (not blurred) images.

Fig. 2 Normal image $\mathbf{I}$ (left); Poisson noised image $\mathbf{I}'$ (right), Canon G9 X Mark II. The mean/standard deviation/median of $|\mathbf{I} - \mathbf{I}'|$ is 5/4/3

Fig. 3 Normal image $\mathbf{I}$ (left); Gaussian blurred image $\mathbf{I}'$ with $\sigma = 0.5$ (right), Nikon D7200. The mean/standard deviation/median of $|\mathbf{I} - \mathbf{I}'|$ is 62/46/53

Gaussian blur Gaussian blurring is a 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 [10]. The filter decreases the contrast between pixels, therefore images are visually more "soft". The experiments were conducted with σ parameter equal 0.001, 0.01, 0.1, 0.2, 0.3 and 0.5. Values of σ parameter close to 0 stand for slight decrease of image quality; values of σ exceeding 0.3 denote strong image quality reduction, what can be seen as Fig. 3. Results of identification for $\sigma = 0.5$ (strongest quality degradation) are collected in Tables 3 and 4.

Results clearly indicate that network recognizes particular models with 99% accuracy for both tested image datasets, which can be considered as almost perfect. This means that even strong image degradation obtained during Gaussian blurring does not prevent from linking the image with the camera. Also the image quality of Gaussian blurred images is not satisfactory, because images (espiecially for high σ values), are strongly blurred.

Random noise Random noise is a technique of image noising, where some pixels are set to distinguishing values (usually 0 or 255 which in RGB model stand for black or white). 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, where k includes 50% of image pixels. An example of such operation is described as Fig. 4. One may assume that replacing such number of pixels will be enough to claim that the image is visually degraded. Results of ISCI identification based on random noised images are presented as Tables 5 and 6.

Fig. 4 Normal image $\mathbf{I}$ (left); random noised image $\mathbf{I}'$ ($k = 50\%$), Nikon D7200. The mean/standard deviation/median of $|\mathbf{I} - \mathbf{I}'|$ is 55/42/46

Fig. 5 Normal image $\mathbf{I}$ (left); LSB-removed image $\mathbf{I}'$ (right), Canon G9 X Mark II. The mean/standard deviation/median of $|\mathbf{I} - \mathbf{I}'|$ is 41/33/36

Table 1 Accuracy ACC = 99% of brand recognition for Dataset I [%], images degraded with Poisson noise

	Ap	Ca	Hu	Ni	Pa	Sa	So	Xi
Apple (Ap)	99	*	*	*	*	*	*	*
Canon (Ca)	*	99	*	*	*	*	*	*
Huawei (Hu)	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	99	*	*	*	*
Panasonic (Pa)	*	*	*	*	99	*	*	*
Samsung (Sa)	*	*	*	*	*	99	*	*
Sony (So)	*	*	*	*	*	*	99	*
Xiaomi (Xi)	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 2 Accuracy ACC = 99% of brand recognition for Dataset II (Dresden Image Database) [%], images degraded with Poisson noise

	Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa (Ag)	99	*	*	*	*	*	*	*	*
Canon (C1)	*	99	*	*	*	*	*	*	*
Casio (C2)	*	*	99	*	*	*	*	*	*
Kodak (Ko)	*	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	*	99	*	*	*	*
Olympus (Ol)	*	*	*	*	*	99	*	*	*
Praktica (Pr)	*	*	*	*	*	*	99	*	*
Rollei (Ro)	*	*	*	*	*	*	*	99	*
Samsung (Sa)	*	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 3 Accuracy ACC = 99% of brand recognition for Dataset I [%], images degraded with Gaussian blur

	Ap	Ca	Hu	Ni	Pa	Sa	So	Xi
Apple (Ap)	99	*	*	*	*	*	*	*
Canon (Ca)	*	99	*	*	*	*	*	*
Huawei (Hu)	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	99	*	*	*	*
Panasonic (Pa)	*	*	*	*	99	*	*	*
Samsung (Sa)	*	*	*	*	*	99	*	*
Sony (So)	*	*	*	*	*	*	99	*
Xiaomi (Xi)	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 4 Accuracy ACC = 99% of brand recognition for Dataset II (Dresden Image Database) [%], images degraded with Gaussian blur

	Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa (Ag)	99	*	*	*	*	*	*	*	*
Canon (C1)	*	99	*	*	*	*	*	*	*
Casio (C2)	*	*	99	*	*	*	*	*	*
Kodak (Ko)	*	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	*	99	*	*	*	*
Olympus (Ol)	*	*	*	*	*	99	*	*	*
Praktica (Pr)	*	*	*	*	*	*	99	*	*
Rollei (Ro)	*	*	*	*	*	*	*	99	*
Samsung (Sa)	*	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 5 Accuracy ACC = 99% of brand recognition for Dataset I [%], adding random noise

	Ap	Ca	Hu	Ni	Pa	Sa	So	Xi
Apple (Ap)	99	*	*	*	*	*	*	*
Canon (Ca)	*	99	*	*	*	*	*	*
Huawei (Hu)	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	99	*	*	*	*
Panasonic (Pa)	*	*	*	*	99	*	*	*
Samsung (Sa)	*	*	*	*	*	99	*	*
Sony (So)	*	*	*	*	*	*	99	*
Xiaomi (Xi)	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 6 Accuracy ACC = 99% of brand recognition for Dataset II (Dresden Image Database) [%], adding random noise

	Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa (Ag)	99	*	*	*	*	*	*	*	*
Canon (C1)	*	99	*	*	*	*	*	*	*
Casio (C2)	*	*	99	*	*	*	*	*	*
Kodak (Ko)	*	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	*	99	*	*	*	*
Olympus (Ol)	*	*	*	*	*	99	*	*	*
Praktica (Pr)	*	*	*	*	*	*	99	*	*
Rollei (Ro)	*	*	*	*	*	*	*	99	*
Samsung (Sa)	*	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Similarly as in case of Poisson noising and Gaussian blurring, the proposed CNN recognizes the devices with high accuracy, even in strong image quality degradation. Therefore, the strategy of randomly noising images cannot be considered as a stable solution to ensure unlinkability between the camera and the image. What is more, the quality of random noised images cannot be considered as satisfactory.

Removing pixels' least significant bit (LSB) Another technique is removing pixels' least significant bit (LSB). More precisely, this denotes setting the least significant bit of pixel intensities for each color channel to 0. We propose to remove the LSB with probability $p = 1.0$. This denotes that all the pixels in the image will be LSB-removed. The sample image obtained during such operation is presented in Fig. 5. Results of brand recognition of LSB-removed images are presented in Tables 7 and 8.

Same as in previous degrading techniques, removing LSB also does not prevent from linking the image with the camera. The classification accuracy of 99% for both image

Table 7 Accuracy ACC = 99% of brand recognition for Dataset I [%], removing LSB

	Ap	Ca	Hu	Ni	Pa	Sa	So	Xi
Apple (Ap)	99	*	*	*	*	*	*	*
Canon (Ca)	*	99	*	*	*	*	*	*
Huawei (Hu)	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	99	*	*	*	*
Panasonic (Pa)	*	*	*	*	99	*	*	*
Samsung (Sa)	*	*	*	*	*	99	*	*
Sony (So)	*	*	*	*	*	*	99	*
Xiaomi (Xi)	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 8 Accuracy ACC = 97% of brand recognition for Dataset II (Dresden Image Database) [%], removing LSB

	Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa (Ag)	99	*	*	*	*	*	*	*	*
Canon (C1)	*	99	*	*	*	*	*	*	*
Casio (C2)	*	*	99	*	*	*	2	*	*
Kodak (Ko)	*	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	*	99	*	*	*	*
Olympus (Ol)	*	*	*	*	*	99	*	*	*
Praktica (Pr)	*	*	*	*	*	*	99	*	*
Rollei (Ro)	*	*	*	*	*	*	*	99	*
Samsung (Sa)	*	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

datasets is still high for both datasets, what is not desired. Moreover, the visual quality of the images is strongly affected, since removing LSB results in reduction of the image luminance by a half.

4.3 Experiment II – Results of individual source camera identification

In this experiment, the results on ISCI identification based on normal JPEG images from tested cameras of both datasets are presented. As mentioned before, in Tables 9 and 10 we present results only on brand recognition.

The accuracy of brand and individual source camera identification is almost perfect for both datasets. The TPRs for all tested cameras are equal 99%.

Table 9 Accuracy ACC = 99% of brand recognition for Dataset I [%]

	Ap	Ca	Hu	Ni	Pa	Sa	So	Xi
Apple (Ap)	99	*	*	*	*	*	*	*
Canon (Ca)	*	99	*	*	*	*	*	*
Huawei (Hu)	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	99	*	*	*	*
Panasonic (Pa)	*	*	*	*	99	*	*	*
Samsung (Sa)	*	*	*	*	*	99	*	*
Sony (So)	*	*	*	*	*	*	99	*
Xiaomi (Xi)	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Table 10 Accuracy ACC = 99% of brand recognition for Dataset II (Dresden Image Database) [%]

	Ag	C1	C2	Ko	Ni	Ol	Pr	Ro	Sa
Agfa (Ag)	99	*	*	*	*	*	*	*	*
Canon (C1)	*	99	*	*	*	*	*	*	*
Casio (C2)	*	*	99	*	*	*	*	*	*
Kodak (Ko)	*	*	*	99	*	*	*	*	*
Nikon (Ni)	*	*	*	*	99	*	*	*	*
Olympus (Ol)	*	*	*	*	*	99	*	*	*
Praktica (Pr)	*	*	*	*	*	*	99	*	*
Rollei (Ro)	*	*	*	*	*	*	*	99	*
Samsung (Sa)	*	*	*	*	*	*	*	*	99

The symbol * denotes values smaller than 1%

Summary Experiments revealed that proposed convolutional neural network almost perfectly recognizes cameras based on images in terms of individual source camera identification (ISCI) aspect. All tested strategies for image quality decreasing which include Poisson noising, Gaussian blurring, adding random noise or removing pixels' least significant bit are not sufficient for breaking the link between the camera and the image, despite significant deterioration of image quality. Analysis of image degradation showed that the proposed network still recognizes devices with high accuracy. Nevertheless, the image quality in all cases of image noising/LSB-removing cannot be considered as satisfactory, what confirms robustness of the proposed CNN to image degrading strategies.

5 Conclusions and future work

In this paper the robustness of digital camera identification with the use of a convolutional neural network (CNN) was discussed. Proposed CNN successfully identifies dozens of cameras based on produced images, as well is robust against image degrading strategies like Poisson noise, Gaussian blur, adding random noise and removing pixels' least significant bit. Extensive experiments conducted on two large image datasets including modern cameras confirmed that camera identification is possible both for normal images as well as for images with quality strongly degraded.

As future work, it is planned to propose an ultra fast method for camera identification, since methods based on convolutional neural networks require high hardware capabilities. Moreover, proposed method should be robust against more sophisticated methods for affecting image quality.

Appendix

We present the Python code (under Keras) for proposed convolutional neural network.

```
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense

num_classes = # define here number of classes
num_epochs = # define here number of epochs

model = Sequential()

model.add(Convolution2D(32, kernel_size=(5, 5), strides=(1, 1),
    ↪ activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Convolution2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Convolution2D(128, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam',
    ↪  metrics=['accuracy'])


from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('dataset/
    ↪ training_set',
                                                 target_size = (64,
                                                     ↪ 64),
                                                 batch_size = 32)

test_set = test_datagen.flow_from_directory('dataset/test_set',
                                            target_size = (64, 64),
                                            batch_size = 32)

H = model.fit(training_set, epochs=num_epochs, verbose=1,
    ↪ validation_data=(test_set))
```

References

1. Agarwal A, Keshari R, Wadhwa M, Vijh M, Parmar C, Singh R, Vatsa M (2019) Iris sensor identification in multi-camera environment. Information Fusion 45:333–345
2. Bernacki J (2021) On robustness of camera identification algorithms. Multim. Tools Appl. 80(1):921–942
3. 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) - Volume 4: SECRYPT, Madrid, Spain, July 24-26, 2017, pp 343–350
4. 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
5. Chen J, Kang X, Liu Y, Wang ZJ (2015) Median filtering forensics based on convolutional neural networks. IEEE Signal Process. Lett. 22(11):1849–1853
6. Chen Y, Huang Y, Ding X (2017) Camera model identification with residual neural network. In: 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017, pp 4337–4341
7. De Marsico M, Nappi M, Riccio D, Wechsler H (2015) Mobile iris challenge evaluation (miche)-i, biometric iris dataset and protocols. Pattern Recogn Lett 57:17–23
8. Debiasi L, Uhl A (2016) Comparison of PRNU enhancement techniques to generate PRNU fingerprints for biometric source sensor attribution. In: 4th International Conference on Biometrics and Forensics, IWBF 2016, Limassol, Cyprus, March 3-4, 2016, pp 1–6
9. documentation M (2020) Add noise to image
10. Flusser J, Farokhi S, IV CH, Suk T, Zitová B, Pedone M (2016) Recognition of images degraded by gaussian blur. IEEE Trans. Image Processing 25(2):790–806
11. Freire-Obregón D, Narducci F, Barra S, Santana MC (2017) Deep learning for source camera identification on mobile devices. CoRR, abs/1710.01257
12. 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
13. Galdi C, Nappi M, Dugelay J-L (2016) Multimodal authentication on smartphones: Combining iris and sensor recognition for a double check of user identity. Pattern Recogn Lett 82:144–153
14. 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
15. Goljan M, Chen M, Comesaña P, Fridrich JJ (2016) Effect of compression on sensor-fingerprint based camera identification. In: Media Watermarking, Security, and Forensics 2016, San Francisco, California, USA, February 14-18, 2016, pp 1–10
16. Goljan M, Fridrich JJ, Chen M (2010) Sensor noise camera identification: countering counter-forensics. In: Memon ND, Dittmann J, Alattar AM, Delp EJ (eds) Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010, Proceedings, vol 7541 SPIE Proceedings, pp 75410S
17. Goljan M, Fridrich JJ, Chen M (2011) Defending against fingerprint-copy attack in sensor-based camera identification. IEEE Trans. Information Forensics and Security 6(1):227–236
18. Goljan M, Fridrich JJ, Filler T (2010) Managing a large database of camera fingerprints. In: Memon ND, Dittmann J, Alattar AM, Delp EJ (eds) Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010, Proceedings, vol 7541 of SPIE Proceedings, pp 754108. SPIE
19. 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

20. Hosseini MDM, Goljan M (2019) Camera identification from HDR images. In: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2019, Paris, France, July 3-5, 2019, pp 69–76
21. Jiang X, Wei S, Zhao R, Zhao Y, Wu X (2016) Camera fingerprint: A new perspective for identifying user's identity. CoRR, abs/1610.07728
22. Kang X, Li Y, Qu Z, Huang J (2012) Enhancing source camera identification performance with a camera reference phase sensor pattern noise. IEEE Trans. Information Forensics and Security 7(2):393–402
23. 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, p 72540K
24. Li C-T, Chang C-Y, 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, pp 19–25
25. Li H, Wang S, Kot AC (2017) Image recapture detection with convolutional and recurrent neural networks. In: Media Watermarking, Security, and Forensics 2017, Burlingame, CA, USA, 29 January 2017 - 2 February 2017, pp 87–91
26. Li R, Li C-T, Guan Y (2018) Inference of a compact representation of sensor fingerprint for source camera identification. Pattern Recogn 74:556–567
27. Lukás J, Fridrich JJ, Goljan M (2006) Digital camera identification from sensor pattern noise. IEEE Trans. Information Forensics and Security 1(2):205–214
28. 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
29. 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
30. Marra F, Poggi G, Sansone C, Verdoliva L (2016) Correlation clustering for prnu-based blind image source identification. In: IEEE international workshop on information forensics and security, WIFS 2016, abu dhabi, united arab emirates, december 4-7, 2016, pp 1–6
31. Marra F, Poggi G, Sansone C, Verdoliva L (2018) A deep learning approach for iris sensor model identification. Pattern Recognit. Lett. 113:46–53
32. Moosavi-Dezfooli S-M, 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
33. 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 2016, saarbrücken, germany, march 21-24, 2016, pp 372–387
34. 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
35. Shaya OA, Yang P, Ni R, Zhao Y, Piva A (2018) A new dataset for source identification of high dynamic range images. Sensors 18(11):3801
36. Taspinar S, Mohanty M, Memon ND (2016) PRNU based source attribution with a collection of seam-carved images. In: 2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016, pp 156–160
37. Thai TH, Retraint F, Cogranne R (2016) Camera model identification based on the generalized noise model in natural images. Digital Signal Processing 48:285–297
38. Tuama A, Comby F, Chaumont M (2016) Camera model identification with the use of deep convolutional neural networks
39. Tuama A, Comby F, Chaumont M (2016) Camera model identification based machine learning approach with high order statistics features. In: 24th European Signal Processing Conference, EUSIPCO 2016, Budapest, Hungary, August 29 - September 2, 2016, pp 1183–1187
40. Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: A survey. Pattern Recogn Lett 119:3–11
41. Yang P, Ni R, Zhao Y (2016) Recapture image forensics based on laplacian convolutional neural networks. In: Digital Forensics and Watermarking - 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers, pp 119–128

42. 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
43. 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