Compact Representation of Digital Camera's Fingerprint with Convolutional Autoencoder

Jarosław Bernacki[a] and Rafał Scherer[b]
Department of Intelligent Computer Systems, Częstochowa University of Technology,
al. Armii Krajowej 36, 42-200 Częstochowa, Poland

Keywords: Digital Camera Identification, Sensor Identification, Digital Forensics, Privacy, Security, Machine Learning, Deep Models, Convolutional Neural Networks.

Abstract: In this paper, we address the challenge of digital camera identification within the realm of digital forensics. While numerous algorithms leveraging camera fingerprints exist, few offer both speed and accuracy, particularly in the context of modern high-resolution digital cameras. Moreover, the storage requirements for these fingerprints, often represented as matrices corresponding to the original image dimensions, pose practical challenges for forensic centers. To tackle these issues, we propose a novel approach utilizing a convolutional autoencoder (AE) to generate compact representations of camera fingerprints. Our method aims to balance accuracy with efficiency, facilitating rapid and reliable identification across a range of cameras and image types. Extensive experimental evaluation demonstrates the effectiveness of our approach, showcasing its potential for practical deployment in forensic scenarios. By providing a streamlined method for camera identification, our work contributes to advancing the capabilities of digital forensic analysis.

1 INTRODUCTION

Digital forensics is a field that has attracted much attention in recent years. One of the most popular topics in digital forensics is the identification of imaging sensors that are present in digital cameras. Nowadays, digital cameras are in general accessible and affordable, which makes them very popular. Smartphones and mobile devices are even more popular. Today's smartphones are equipped with built-in digital cameras which encourage people to take photos and share them on social media networks. However, the possibility of establishing whether an image was taken by a given camera may expose users' privacy to a serious threat. Hence, a number of papers in recent years are dedicated to the study of imaging device artifacts that may be used for digital camera identification.

Digital camera identification can be realized in two approaches: individual source camera identification (ISCI) and source model camera identification (SCMI). The ISCI is capable of distinguishing a certain camera model among cameras of both the same and different camera models. On the other hand, the SCMI distinguishes a certain camera model among the different models but is not able to distinguish a certain copy of a camera from other cameras of the same model. For instance, if we have the following cameras: Canon EOS R (0), Canon EOS R (1), ..., Canon EOS R (n), Nikon D780 (0), Nikon D780 (1), Sony A1 (0), Sony A1 (1), the ISCI will distinguish all cameras as different. The SCMI would distinguish only the general models, i.e. Canon EOS R, Nikon D780, and Sony A1. Therefore, it is the limitation of the SCMI approach. This motivates to develop such methods and algorithms for camera identification that they would work in terms of the ISCI aspect.

The state-of-the-art algorithm for the ISCI aspect was proposed by Lukás et al.'s (Lukás et al., 2006). This algorithm used a so-called photo response non-uniformity (PRNU) that is present in images and allows for camera identification. The PRNU $\mathbf{N}$ may be calculated in the following manner: $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$, where $\mathbf{I}$ is an input image and F is a denoising filter. The PRNU serves as a unique camera's fingerprint. Many studies (Bruno et al., 2020; Mandelli et al., 2020; Picetti et al., 2020) confirmed the high efficacy of camera identification in such a way. However, this approach shows some weaknesses. The greatest disadvantage is the representation of the camera's fingerprint which is represented as a matrix in the original

[a] https://orcid.org/0000-0002-4488-3488
[b] https://orcid.org/0000-0001-9592-262X

Bernacki, J. and Scherer, R.
Compact Representation of Digital Camera's Fingerprint with Convolutional Autoencoder.
DOI: 10.5220/0012821300003767
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 792-797
ISBN: 978-989-758-709-2; ISSN: 2184-7711
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

images' dimensions. This may be problematic in the aspect of storing a large number of PRNUs in some forensic centers. This motivates to work out a method that will minimize this problem.

1.1 Contribution

In this paper, we propose a method that uses a convolutional autoencoder (AE) to generate a compact (compressed) representation of the camera's fingerprint. This compact representation of the camera's fingerprint may be successfully used to perform individual source camera identification with comparable accuracy to state-of-the-art methods. However, our method solves the problem connected with storing large cameras' fingerprints, since the representation of the fingerprint is not stored as a matrix, but as a vector. We conduct experiments on a large set of modern digital cameras which confirm the similar accuracy with state-of-the-art methods.

1.2 Organization of the Paper

The paper is organized as follows. The next section discusses previous and related works. In Section 3 the problem is formulated and the proposed method is described with a brief recall of state-of-the-art methods. Section 4 presents results of classification compared with literature methods. The final section concludes this work.

2 PREVIOUS AND RELATED WORK

In (Valsesia et al., 2015) there is presented an algorithm for the camera's fingerprint compact representation. For this purpose, a random projection matrix whose dimensions will be matched to the camera's fingerprint matrix in terms of matrix multiplication must be generated. The random projected matrix is then multiplied by the camera's fingerprint matrix which produces a new matrix. Such a new matrix serves as a compact camera's fingerprint representation and is much lower than the original fingerprint. The accuracy of this method is similar to the use of original fingerprints, which makes the considered method useful. However, such an approach requires generating random matrix and matrix multiplication, which may not be computationally optimal. A linear discriminant analysis used to extract more discriminant sensor pattern noise (SPN) features is discussed (Li et al., 2018). The compact representation of the SPN is featured as a vector.

In (Liu et al., 2021) a patch-level camera identification with the convolutional neural networks (CNN) is described. The advantage of the method is also image tampering detection. In (Cozzolino et al., 2021) a generative adversarial network (GAN) for compromising the PRNUs is presented. Considered GAN produces synthetic images that are injected with other cameras' traces. Experiments confirmed that GAN-generated images may successfully deceive state-of-the-art algorithms for camera identification. In (Lai et al., 2021) a Hierarchy Clustering method for the camera's fingerprint identification is discussed. Such a novel approach allows for the classification without training image datasets. In (Salazar et al., 2021) there is proposed a method for clustering the cameras' fingerprints. The images are distinguished by applying various denoising algorithms. In (Borole and Kolhe, 2021) a fuzzy min-max neural network is considered for the identification of the camera's digital fingerprint (PRNU). The PRNU patterns are represented as Hu's invariants and then passed into a neural network for training and classification. The experimental evaluation confirmed the accuracy of the proposed method. In (Rafi et al., 2021) a PRNU-based method for camera identification is described. This is realized with the use of a convolutional neural network that is adopted to eliminate scenes in the images that obscure the noise used to calculate the PRNU. Experiments confirmed that considered methods achieve high accuracy.

3 CAMERA'S FINGERPRINT COMPACT REPRESENTATION WITH CONVOLUTIONAL AUTOENCODER

3.1 Problem Description

Many papers are focused only on high classification accuracy methods for digital camera identification and generally do not consider the aspect of compact representation of cameras' fingerprints. The identification based on camera's fingerprint $\mathbf{N}$ is calculated with the formula presented as Eq. 1 (Lukás et al., 2006; Tuama et al., 2016):

$$\mathbf{N} = \mathbf{I} - F(\mathbf{I}) \tag{1}$$

where $\mathbf{I}$ is the input image, F stands for a denoising filter. It should be mentioned that the $\mathbf{N}$ is calculated only for one image of a particular camera. This procedure should be repeated for a certain number of images (at least 45 (Lukás et al., 2006)) to calculate the final camera's fingerprint. The procedure presented as Eq. 1 denotes that the size of the fingerprint **N** is equal to the size of the input image **I**. Worth mentioning that for modern cameras producing large dimensions images (for example 6000×4000 or 7000×5000 pixels) stored in forensics centres such fingerprints may be problematic. Therefore, we propose a method utilizing a convolutional autoencoder (AE) that may be useful to fill this gap. Convolutional autoencoders are widely used for different problems, including dimensionality reduction, anomaly detection, generating new features, recommender systems, and many more. We propose to use the AE in terms of reducing dimensionality. In our approach, the fingerprint **N** will be reduced into a much smaller representation than an input image **I**.

To obtain the autoencoder learning the specificity of the camera (not the content of the input image), it is essential to denoise the cameras' images. We use the well-known formula presented as Eq. 1, utilized in (Lukás et al., 2006; Tuama et al., 2016) to calculate the residuum **N**. The **N** images are passed to the input of the autoencoder with the label of the cameras that a particular residuum comes from. Then, the autoencoder calculates the latent vector, which we find as a compact representation of **N**. For the classification purposes we are not interested with the typical decoding part of the autoencoder. Therefore, such operation provides us the compact representation of the residuum **N**.

3.2 State-of-the-Art: Existing CNNs

For evaluation, we refer to some convolutional neural network-based methods which include: Mandelli et al.'s (Mandelli et al., 2020) and Kirchner & Johnson (Kirchner and Johnson, 2020). Let us briefly recall the structure of Mandelli et al.'s convolutional neural network (CNN):

(1) A first convolutional layer of kernel 3×3 producing feature maps of size 16×16 pixels with Leaky ReLU as an activation method and max-pooling;

(2) A second convolutional layer of kernel 5×5 producing feature maps of size 64×64 pixels with Leaky ReLU as an activation method and max-pooling;

(3) A third convolutional layer of kernel 5×5 producing feature maps of size 64×64 pixels with Leaky ReLU as an activation method and max-pooling;

(4) A pairwise correlation pooling layer;

(5) Fully connected layers.

For more details related to the structure of the network, we refer to the authors' paper, due to paper limitations.

3.3 Proposed Method: Convolutional Autoencoder

As mentioned, we propose to use the convolutional autoencoder to reduce the dimensionality of cameras' fingerprints. The method aims to take the residuum **N** (calculated as Eq. 1) and produce its compact representation by using the autoencoder. For this purpose, we use only the encoding part of the autoencoder – the decoding part may be skipped since we do not need it. The structure of the proposed convolutional autoencoder is defined as follows:

(1) A first convolutional layer of 64 filters of size 3×3 (stride 2), with ReLU as an activation function, followed by a Max-Pooling layer + padding 1;

(2) A second convolutional layer of 32 filters of size 3×3 (stride 2), with ReLU as an activation function, followed by a Max-Pooling layer + padding 1;

(3) A third convolutional layer of 16 filters of size 3×3 (stride 2), with ReLU as an activation function, followed by a Max-Pooling layer + padding 1;

The activation function for the autoencoder is sigmoid. We assume to process images of size 128×128. Therefore, the size of the latent vector, storing the compact representation, may be calculated in the following manner:

(1) After the first convolutional layer, the feature map size is $(128-3+2\cdot 1)/2+1 = 64 \times 64$;

(2) After the second convolutional layer, the feature map size is $(64-3+2\cdot 1)/2+1 = 32 \times 32$;

(3) After the third convolutional layer, the feature map size is $(32-3+2\cdot 1)/2+1 = 16 \times 16$.

Since the number of channels in the last feature map is 16 and its spatial dimensions equal 16×16, the size of the latent vector is $16 \cdot 16 \cdot 16 = 4096$.

The latent vector obtained in the described procedure is considered as a compressed camera's fingerprint. It may be used for classification purposes both for using CNN-based classifiers, as well as with classic machine learning algorithms. Worth mentioning that the latent vector may be stored as a single text file which is efficient in terms of disk and hardware usage.

The Discriminator. To perform the classification, we propose to use the discriminator. The idea of the

discriminator is similar to the Generative Adversarial Network (GAN) (Goodfellow et al., 2014). The use of the discriminator is essential because the autoencoder's latent vector containing the compact representation of the residuum **N** should be passed into the classifier. The discriminator may be realized with the standard convolutional neural network, however, one may use well-known machine learning algorithms, such as Support Vector Machine (SVM).
The structure of the sample discriminator is described below:

(1) Latent vector of the autoencoder + camera ID (label);

(2) A first convolutional layer of 32 filters of size 3×3 with ReLU as an activation function, stride 2, followed by a max-pooling layer;

(3) A second convolutional layer of 64 filters of size 3×3 with ReLU as an activation function, stride 2, followed by a max-pooling layer;

(4) A third convolutional layer of 128 filters of size 3×3 with ReLU as an activation function, stride 2, followed by a max-pooling layer;

(5) Fully connected 512 + dropout 0.5 + ReLU;

(6) Fully connected 128 + dropout 0.5 + ReLU.

The activation function is softmax.

All meta-parameters both for the proposed autoencoder, as well the discriminator were determined experimentally.

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setup and Preliminaries

We compare the efficacy of identification of particular cameras both by proposed convolutional autoencoder (AE) and the following state-of-the-art methods: by Mandelli et al.'s (more details about Mandelli's method are presented in Subsec. 3.2), CNN by Kirchner & Johnson (Kirchner and Johnson, 2020), Lukás et al.'s algorithm (Lukás et al., 2006), Valsesia et al.'s algorithm (Valsesia et al., 2015) and Li et al.'s algorithm (Li et al., 2018). Both proposed AE and CNNs are learned by 100 epochs. Worth mentioning that methods presented by Valsesia and Li generate compressed camera fingerprint representations by their procedures. Due to paper limitations, we refer to the mentioned authors' papers to get acquainted with cited algorithms.

We use a set of more than 60 modern cameras (Bernacki and Scherer, 2023). The used cameras include (i.a.): Canon EOS 1D X Mark II (C1), Canon EOS 5D Mark IV (C2), Canon EOS 90D (C3), Canon EOS M5 (C4), Canon EOS M50 (C5), Canon EOS R (C6), Canon EOS R6 (C7), Canon EOS RP (C8), Fujifilm X-T200 (F1), Nikon D5 (N1), Nikon D6 (N2), Nikon D500 (N3), Nikon D780 (N4), Nikon D850 (N5), Nikon Z6 (N6), Nikon Z6 II (N7), Nikon Z7 (N8), Nikon Z7 II (N9), Sony A1 (S1), Sony A9 (S2). At least 40 images per camera are used for learning.

As evaluation, we use standard *accuracy* (ACC) measure, defined as:

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

where TP/TN denotes "true positive/true negative"; FP/FN stands for "false positive/false negative". TP denotes the number of cases correctly classified to a specific class; TN refers to instances that are correctly rejected. FP denotes cases incorrectly classified to the specific class; FN is cases incorrectly rejected.

Experiments are held on a notebook Gigabyte Aero equipped with the Intel Core i7-13700H CPU with 32 gigabytes of RAM and Nvidia GeForce RTX 4070 GPU with 8 gigabytes of video memory. Scripts for the proposed convolutional autoencoder and state-of-the-art CNNs are implemented in Python under the PyTorch framework with Nvidia CUDA support.

4.2 Results of Classification

Due to paper limitations, we do not present the results of the cameras' classification as confusion matrices.

Results showed that identification reaches a similar efficacy, in both of the proposed autoencoder and state-of-the-art procedures. All methods obtain the average identification accuracy at 91-95% which we may find satisfactory. In detail, the classification of the compressed representation of fingerprints generated by the proposed AE obtains 95% accuracy. CNNs presented by Mandelli et al.'s and Kirchner & Johnson also achieve 95%. The Lukás et al.'s algorithm also reaches 95%, while Valsesia and Li et al.'s reach a bit lower results, 92% and 91%, respectively. This means that camera identification based on the latent vector of the proposed autoencoder is as accurate as state-of-the-art methods.

One may assume that proposed AE and CNN-based methods would produce higher classification accuracy, if their structure was deeper. However, results indicate that the proposed AE does not stand out from the literature's methods.

4.3 Time Performance

Speed of Learning. We have compared the time needed for learning the proposed convolutional autoencoder and CNN-based methods. Results may be seen in Fig. 1. Results indicate that learning the proposed convolutional autoencoder requires less time per epoch than using state-of-the-art CNNs. One epoch using the proposed AE is passed over 0.1 of a minute while using CNN turns to about 0.3 of a minute. Therefore, the overall time for passing 100 epochs requires about 8 minutes for the proposed AE and at least 15 minutes for CNNs. Thus, it confirms the twice advantages over the literature methods. The number of 100 epochs is necessary to obtain the identification accuracy at the level about 95% per device.

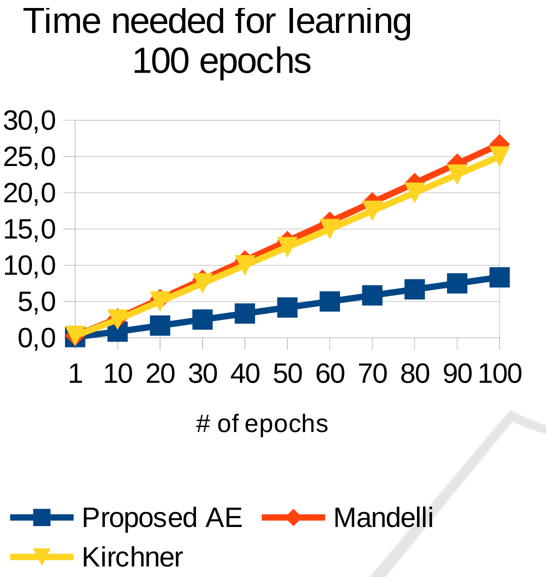

Figure 1: Comparison of time needed for learning 100 epochs.

Fingerprint Size. We have compared the fingerprint weight both of the proposed autoencoder and Lukás et al.'s algorithm. Results indicated that the latent vector generated by the proposed autoencoder (which as mentioned before we treat as the camera's fingerprint) is much smaller in terms of file weight than fingerprints generated with Lukás et al.'s algorithm. A text file storing a latent vector requires about 3-4 megabytes, while Lukás file representing a matrix may weigh even about 120 megabytes. Decreasing fingerprint weight may play a crucial role for forensics centers that store such materials.

Calculating Compact Representations. In Fig. 2 we describe the time that is required to generate compressed representations of cameras' fingerprints by Valsesia and Li et al.'s algorithms.

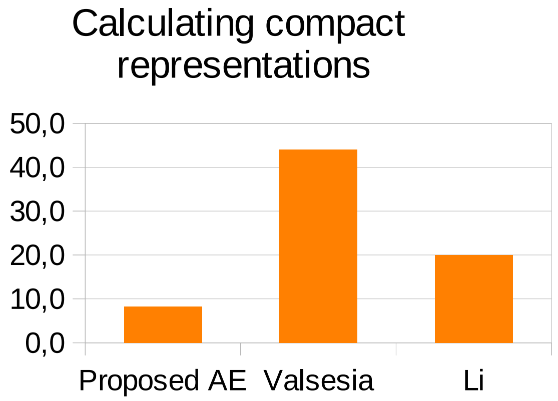

Figure 2: Comparison of time needed for generating compressed fingerprint representations.

Results point out that Valsesia and Li methods require more time to generate their compressed representations of fingerprints. The proposed AE needs about 8-9 minutes to generate a latent vector based on 40 images, while the Valsesia and Li methods require 44 and 20 minutes, respectively. Thus, the proposed method obtains better time performance than the considered methods from the literature.

Number of Epochs. We have analyzed, how classification accuracy increases with the number of training epochs. Intuitively, the more number of epochs, the more classification accuracy. Results are presented in Fig. 3.

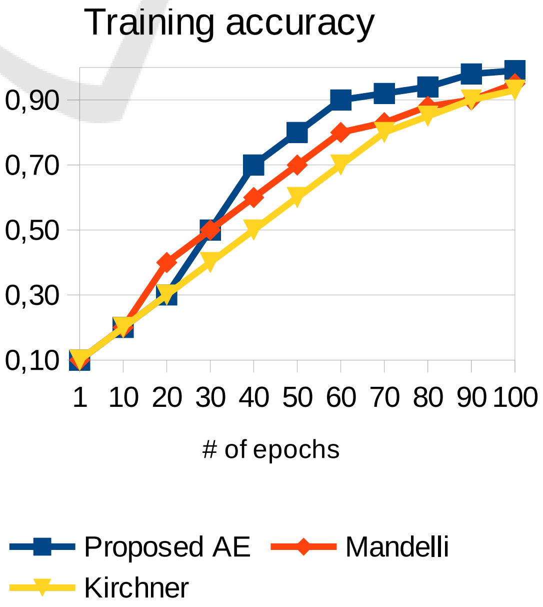

Figure 3: Comparison of training accuracy.

The proposed AE requires a similar number of training epochs to obtain comparable classification accuracy as state-of-the-art CNNs. To achieve 90% training accuracy, all methods need to be learned by at least 85 epochs. Such a number of epochs is sufficient to obtain about 90% identification accuracy. However, training further for 100 epochs allows for identification accuracy, as presented in the previous section. This means that our proposed architecture does not suffer both from training and identification accuracy, compared to existing methods.

5 CONCLUSION

In this paper, we have proposed a method for individual source camera identification based on cameras' fingerprints. The solution was based on a convolutional autoencoder which was used to produce a compact representation of cameras' fingerprints. Extensive experimental evaluation conducted on a large number of modern imaging devices and enhanced with a statistical analysis confirmed the reliability of the proposed method. Convolutional autoencoder-based digital camera identification was realized with high identification accuracy. The great advantage of the proposed method is the possibility of storing cameras' fingerprints in a compact representation, which may aim forensic centers to save space for storing such fingerprints.

As future work, we consider a solution utilizing multiple convolutional autoencoders. One may consider a scenario utilizing one convolutional autoencoder per each camera which would be a useful foundation for anomaly detection.

REFERENCES

Bernacki, J. and Scherer, R. (2023). Imagine dataset: Digital camera identification image benchmarking dataset. In *Proc. 20th Int. Conf. Security and Cryptography—SECRYPT*, pages 799–804. INSTICC, SciTePress.

Borole, M. and Kolhe, S. R. (2021). A feature-based approach for digital camera identification using photo-response non-uniformity noise. *Int. J. Comput. Vis. Robotics*, 11(4):374–384.

Bruno, A., Cattaneo, G., and Capasso, P. (2020). On the reliability of the pnu for source camera identification tasks. *arXiv preprint arXiv:2008.12700*.

Cozzolino, D., Thies, J., Rossler, A., Nießner, M., and Verdoliva, L. (2021). Spoc: Spoofing camera fingerprints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 990–1000.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Kirchner, M. and Johnson, C. (2020). SPN-CNN: boosting sensor-based source camera attribution with deep learning. *CoRR*, abs/2002.02927.

Lai, Z., Wang, Y., Sun, W., and Zhang, P. (2021). Automatic source camera identification technique based-on hierarchy clustering method. In Sun, X., Zhang, X., Xia, Z., and Bertino, E., editors, *Artificial Intelligence and Security - 7th International Conference, ICAIS 2021, Dublin, Ireland, July 19-23, 2021, Proceedings, Part II*, volume 12737 of *Lecture Notes in Computer Science*, pages 715–723. Springer.

Li, R., Li, C., and Guan, Y. (2018). Inference of a compact representation of sensor fingerprint for source camera identification. *Pattern Recognition*, 74:556–567.

Liu, Y., Zou, Z., Yang, Y., Law, B. N., and Bharath, A. A. (2021). Efficient source camera identification with diversity-enhanced patch selection and deep residual prediction. *Sensors*, 21(14):4701.

Lukás, J., Fridrich, J. J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Trans. Information Forensics and Security*, 1(2):205–214.

Mandelli, S., Cozzolino, D., Bestagini, P., Verdoliva, L., and Tubaro, S. (2020). Cnn-based fast source device identification. *IEEE Signal Processing Letters*, 27:1285–1289.

Picetti, F., Mandelli, S., Bestagini, P., Lipari, V., and Tubaro, S. (2020). DIPPAS: A deep image prior PRNU anonymization scheme. *CoRR*, abs/2012.03581.

Rafi, A. M., Tonmoy, T. I., Kamal, U., Wu, Q. M. J., and Hasan, M. K. (2021). Remnet: remnant convolutional neural network for camera model identification. *Neural Comput. Appl.*, 33(8):3655–3670.

Salazar, D. A., Ramirez-Rodriguez, A. E., Nakano, M., Cedillo-Hernandez, M., and Perez-Meana, H. (2021). Evaluation of denoising algorithms for source camera linking. In *Mexican Conference on Pattern Recognition*, pages 282–291. Springer.

Tuama, A., Comby, F., and Chaumont, M. (2016). Camera model identification with the use of deep convolutional neural networks. In *IEEE International Workshop on Information Forensics and Security, WIFS 2016, Abu Dhabi, United Arab Emirates, December 4-7, 2016*, pages 1–6. IEEE.

Valsesia, D., Coluccia, G., Bianchi, T., and Magli, E. (2015). Compressed fingerprint matching and camera identification via random projections. *IEEE Transactions on Information Forensics and Security*, 10(7):1472–1485.