AN AUTO-ENCODER BASED METHOD FOR CAMERA FINGERPRINT COMPRESSION

Kaixuan Zhang, Zihan Liu, Jiashang Hu, Shilin Wang, Senior Member, IEEE*

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

ABSTRACT

Camera fingerprint links a picture to its camera sensor, which is widely applied in sensor device identification, social network tracing and forgery detection. However, such fingerprints are in high dimensionality and cost substantial memory and computing resources, limiting their uses in real-time processing on embedded devices. In this paper, we introduce a new method to compress high-dimensional floating-point fingerprints to low-dimensional binary features to save storage as well as maintaining their representative abilities. Also, we present a much faster approach to sensor device matching with hamming distance, compared with the commonly used Peak to Correlation Energy (PCE) distance. Our method contains two stages. First, raw fingerprints are compressed into low-dimensional features with our proposed grouping strategy and auto-encoder based model. Then, the compressed floating-point features are further converted into more compact binary features. Experiments show that our method achieves superior performance over several competitive compression methods in both identification and verification tasks.

***Index Terms*—** Camera Fingerprint, Compression, Binarization, Auto-encoder

1. INTRODUCTION

Sensor identification is an important topic in the field of digital forensics. For the purposes of copyright protection and malicious photo publishers tracking, it is necessary to develop effective and efficient approaches to tracing the sources of photos online. Although some meta information like file headers can be used to identify the sensor, this kind of information can be easily stripped off or modified [?].

Lukás et al. [?] finds that the device leaves a unique texture mark on the pictures among the pixels, which can be extracted as device "fingerprint". The method, also known as Photo Response Non-Unit (PRNU), is the most common approach to tracing the image source. The uniqueness of the camera fingerprints are proved in researches [?, ?]. Besides the conventional method, several PRNU extraction methods have been proposed to improve efficiency [?, ?]. Other researchers focus on improving the accuracy of PRNU. Tiwari et al. [?] designs a weighting function to balance the information from different parts of the image with different image quality. Zeng et al. [?] adopts wavelet transform to extract the more precise noise patterns from images. Mandelli et al. [?] uses Convolutional Neural Network to make more precise predictions. Except for source sensor identification, PRNU can also be used in forgery detection [?] and social network tracing [?]. Although PRNU works well in many situations, the dimension of the extracted fingerprint is equal to or proportional with the size of the original image. Therefore, high resolution photos nowadays lead to large fingerprints, costing substantial memory and computing resources [?].

To reduce the storage cost and speed up the matching process, many approaches have been proposed. Goljan et al. [?] uses the digest of the fingerprints rather than the original noise, which reduces the computational overhead to some degree. Bayram et al. [?] designs an approach to compress the sensor fingerprint into binary-quantized form, and proves that the hamming distance and the Peak to Correlation Energy (PCE) distance are related so that the performance can be preserved after binarization. Li et al. [?] adopts Principle Component Analysis (PCA) to compress the fingerprint representation. With theoretical support of [?], Random Projection method is utilized to project fingerprint feature from high-dimensional to low-dimensional space [?, ?, ?, ?, ?]. Bernacki et al. [?] proposes a method called CompaRe, which divides PRNU feature into submatrices and uses trace of them as representation.

The existing approaches are able to compress the sensor fingerprint to some extent. However, there are some drawbacks in each method. The spatial relations between a pixel and its neighbors are not modeled in digest [?] method. In [?], non-diagonal pixels in each submatrices are simply dropped when compressing. [?] approximates large matrix multiplication with Fast Fourier Transform (FFT), which leads to the loss of compact features' representative ability. The PCA based method [?] requires substantial memory. In this paper, we propose a novel auto-encoder based approach to compressing the PRNU fingerprints into low-dimensional binary features. It utilizes all information in original features, models the spatial relations between pixels and their neighbors, and provides a flexible framework to balance reconstruction and

*Shilin Wang is the corresponding author. This work was supported by the National Natural Science Foundation of China (62271307, 61771310) and Key RD Program Major (Key) Project of Science and Technology Department of Ningxia.

discrimination ability. Experiments on the IMAGINE public dataset indicate better performance on downstream tasks are achieved by our method.

2. PROPOSED METHODS

2.1. Motivation

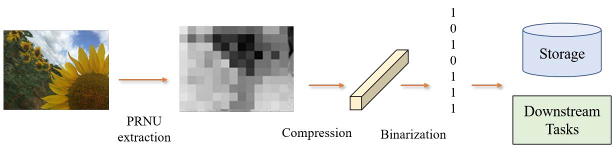

Fig. 1. The Overall Pipeline of the Designed Method

As illustrated in Fig.1, the proposed pipeline includes two stages: compression and binarization. Given a picture I, the PRNU noise feature F is extracted via PRNU extraction method [**?**]. In compression stage, an auto-encoder based network is introduced. Besides, a grouping strategy is applied to reduce model complexity. The PRNU noise feature is divided into multiple patches and compressed separately. In binarization stage, a method utilizing average value is adopted. The binary feature can be stored on embedded devices and applied to downstream tasks such as device linking and device identification.

Compressed features are supposed to have representative abilities in variant downstream tasks. Different from supervised learning from device labels in [**?**], we expect our model to have good generalization in all devices. Therefore, we use unsupervised learning method to build our model. Auto-encoder plays a fundamental role in unsupervised learning and compression [**?**, **?**]. There is few research on compressing PRNU with auto-encoder. We utilize auto-encoder to model spatial relations for most pixels.

PCA based methods also models relations among pixels [**?**], but such method require lots of memory. We aims to design a lightweight model to facilitate applications on embedded devices. Therefore, a grouping scheme is adopted, which reduces calculation costs by n times (n refers to blocks number). Orthogonality regularization [**?**] has been introduced to achieve better accuracy and stable convergence. In our model, orthogonal restriction on parameters boosts discrimination abilities in downstream tasks. We also provide a more flexible approach to optimize training parameters, where norm type and orthogonality weight are hyperparameters, so that reconstruction and discrimination can be balanced.

2.2. Compression

In compression stage, a PRNU feature $F \in \boldsymbol{R}^{H\times W}$ is supposed to be compressed to the dense feature $F^D \in \boldsymbol{R}^m$. The

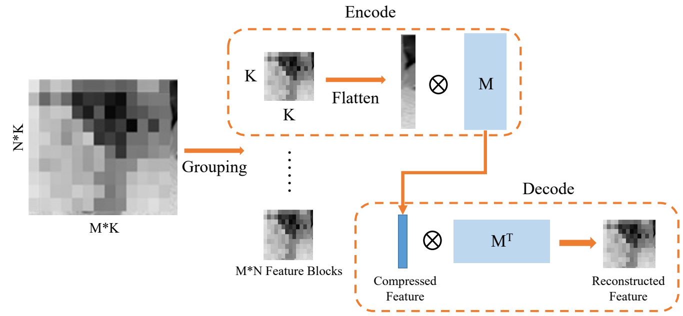

Fig. 2. Illustration of the compression process. The compressed feature is extracted using an auto-encoder based network and grouping strategy.

details of the auto-encoder based network are depicted in Fig. 2. Suppose the original feature F is a two-dimensional matrix of length H and width W with floating-point value. First, the feature is divided into $n = HW/k^2$ blocks. Hereafter we denote i-th block as F_i, and $i = 0, 1, \ldots, n-1$. Each block $F_i \in \boldsymbol{R}^{k\times k}$ will be flattened into $F_i^f \in \boldsymbol{R}^{k^2}$, then compressed separately. Finally, the compressed fingerprint is obtained by concatenation of hidden features of all blocks.

An orthogonal matrix $M_i \in \boldsymbol{R}^{k^2\times z}$ is introduced for each block to map the k^2 sized feature into a compact code sized z, where $z = m/k^2$, as illustrated in Eq. 1. The matrix works as an encoder to generate the hidden feature $F_i^E \in \boldsymbol{R}^m$. Meanwhile, the transposed matrix of M_i works as the decoder. By multiplying M_i^T to hidden feature, the reconstructed feature $F_i^D \in \boldsymbol{R}^{k^2}$ can be obtained.

$$\begin{aligned} F_i^E &= F_i^f M \\ F_i^D &= F_i^E M^T \end{aligned} \tag{1}$$

Two losses are added to train the orthogonal matrix M. First, to preserve the property of the original fingerprint, it is necessary to ensure that the reconstructed feature and the original feature are as similar as possible, thus reconstruction loss L_r is used.

Second, to ensure the symmetry between encoder and decoder, the orthogonality of the matrix M_i needs to be guaranteed. Orthogonality of M_i also limits the scale of the hidden feature and prevents signal vanishing or explosion [**?**]. Therefore in orthogonality loss L_o, the multiplication result of M_i and M_i^T is supposed to be close to unit matrix E.

The whole loss function is illustrated in Eq. 2. By optimizing the loss function, M is trained to preserve robust property to hidden feature. Meanwhile, the hyper-parameter λ is introduced to balance reconstruction and discrimination.

$$\begin{aligned} L_r &= \sum_{i=0}^{n-1} \|F_i^D - F_i^f\|_{norm} \\ L_o &= \sum_{i=0}^{n-1} \|E - M_i M_i^T\|_{norm} \\ L &= L_r + \lambda * L_o \end{aligned} \tag{2}$$

Norm in Eq. 2 is a hyperparameter which can be assigned to L1 norm or L2 norm to measure the distance of two matrices (or features). L1 norm is adopted in our experiment. The SGD optimizer is applied to train the compression network. The initial learning rate is set to 100. It is relatively high because we discover in practice that gradients in M_i are small. The network is trained for 10,000 iterations at batch size of 16. λ is set to 10^2 to balance magnitude of the two loss functions.

For the inference phase, the PRNU feature is first grouped into blocks of size k. Then each block is flatten and projected via M_i to a compact representation of size z. By concatenating the compressed features from each block, the fingerprint compression process is completed.

2.3. Binarization

The binarization stage converts compressed feature $F^D \in \boldsymbol{R}^m$ to binary feature $B = \{0,1\}^m$, which is the final compact representation of PRNU feature. Bayram et al. [?] have found correlations between hamming distance and PCE distance, thus we follow their simple yet effective binarization method, as shown in Eq. 3. $\bar{x} \in R$ denotes average value of feature $x \in \boldsymbol{R}^m$.

$$B_i = \begin{cases} 0 & if \quad x_i < \bar{x} \\ 1 & if \quad x_i >= \bar{x} \end{cases} \tag{3}$$

3. EXPERIMENTS

3.1. Experiment Setup

IMAGINE dataset [?] is employed to evaluate the performance of the proposed method. It consists of images taken from 66 devices. An open-set protocol is adopted. The dataset is divided into training set (composed of 46 devices) and test set (composed of 20 devices). Results of all experiments are average value on 5 different train-test splits. The extraction process strictly follows Luca Bondi et al.[1]

The compression and binarization processes are supposed to maintain the PRNU's discrimination among different devices. Therefore, our model is evaluated in two tasks, as Fig. 3 shows. In identification task, PRNU prototypes are generated on 15 photos for each device in test set. In verification task, we select 50 positive samples and 50 negative samples for each sample to generate test pairs.

Central $P \times P$ pixel region of original images are cropped (P is referred as image size in the tables below). We tested the efficiency of our method by two tasks below. **Identification task**: given an image and PRNUs of n devices, identify which device acquired the image. It is also called the device identification problem in [?]. **Verification task**: given a pair of images, predict whether they were acquired from the same device. It is also called the device linking problem in [?].

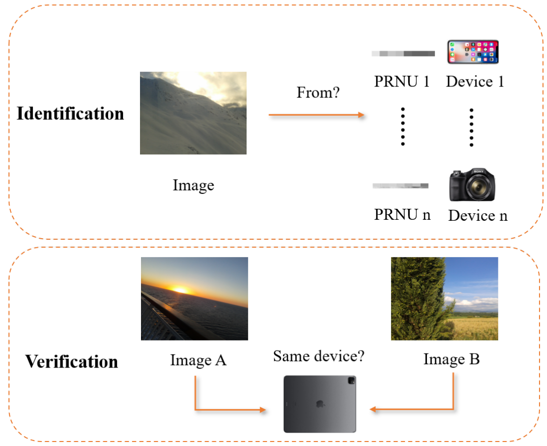

Fig. 3. Two tasks in our experiments: Identification and Verification.

For the identification problem, the predicted device is supposed to have the closest compact representation to the query's, and the accuracy of predictions is adopted as evaluation metric. For the verification problem, we calculate the distance between compact representation of two query PRNUs and adopt AUC as evaluation metric.

3.2. Identification and Verification Performance Analysis

The experiments on PRNU fingerprints compression are conducted. We choose the most popular three methods digest [?], Random Projection [?] and CompaRe [?] to compare with. Results in two tasks and two cropping size are listed in Table 1.

Table 1. Model Performance on Camera Identification and Verification

Image Size	720		1080	
Task	ID	Ver	ID	Ver
Origin	71.78	73.39	76.74	82.93
Digest [?]	56.81	71.10	63.10	76.61
RP [?]	47.63	66.91	57.62	72.30
CompRe [?]	58.33	72.77	65.55	78.42
Ours	**62.38**	**75.58**	**66.73**	**80.40**
Image Size	1800		3600	
Task	ID	Ver	ID	Ver
Digest [?]	67.60	82.24	80.24	85.59
RP [?]	64.27	80.19	77.93	82.28
CompRe [?]	69.37	**84.49**	81.00	84.58
Ours	**70.94**	84.22	**81.84**	**85.86**

Note that the result of "Origin" method shows the discrimination of original PRNU features, where the similarity is calculated via PCE. And four methods below are evaluated at compression rate at 16, where $d = P \times P$ and refers to

[1]https://github.com/polimi-ispl/prnu-python.git

the size of cropped images. Table 1 indicates that our auto-encoder based method has achieved the best performance in most cases.

Considering realistic scenarios, time and space consumption are also noted in verification task of image size 720. The comparison result is demonstrated in Table 2. For four compression methods, T_1 (compression time, ms/sample) refers to the duration of compression and binarization, and T_2 (distance time, ms/sample) refers to duration of calculating distance between query pairs. Suppose original PRNU feature is stored in format of float32. Fast Fourier Transform (FFT) in RP and inference of our methods are calculated via GPU.

Table 2. Time and Space Consumption in Verification Task

	AUC	T_1	T_2	Storage
Origin	73.39	0	90.91	1.98MB
Digest [?]	71.10	2.42	0.04	31.64KB
RP [?]	66.91	3.79		
CompRe [?]	72.77	4.03		
Ours	75.58	4.60		

Table 2 shows the trade off between performance and consumption. Our compression method involved with large amounts of linear projections, which takes marginally longer time when compared to former ones.

3.3. Hyperparameter Analysis

Orthogonal Loss L_o **.** Ablation study on orthogonal loss L_o is conducted. Results are shown in Table 3. Orthogonal loss is essential in our method, which force the independence of the compressed feature.

Table 3. Ablation Study on Orthogonal Loss Weight

Image Size	720		1080	
Task	ID	Ver	ID	Ver
w/o L_o	58.58	73.11	62.90	79.35
w/ L_o	62.38	75.58	66.73	80.40

Block Size n **.** Our method requires dividing noises to small square blocks. Therefore, block size is a crucial hyperparameter. To discover its robustness, experiments on various block sizes at compression rate of 16 are conducted. Results of two tasks and two cropping size are in Table 4.

Table 4. Performance at Different Block Size

Image Size	720		1080	
Task	ID	Ver	ID	Ver
Block size = 4	62.38	75.58	66.73	80.40
Block size = 8	60.10	75.12	66.14	79.99
Block size = 16	59.02	74.32	-	-

It can be observed that smaller block performs slightly better than larger block. This is mainly because trainable parameters in smaller block experiment are less than larger ones. The amounts of samples is limited in IMAGINE dataset, so simpler model is easier to optimize. However, the identification and verification results do not strongly depend on the choice of block size. Therefore, small block size is recommended due to its better performance and less parameters.

Compression Rate r **.** To meet different storage budgets, model performance at different compression rate $r = d/m$ of binary features are conducted at block size of 8, where $r = 4, 16, 64$. Fig. 4 shows the trade off between bit length and performance. Note that the legend below refers to compression methods and cropping size.

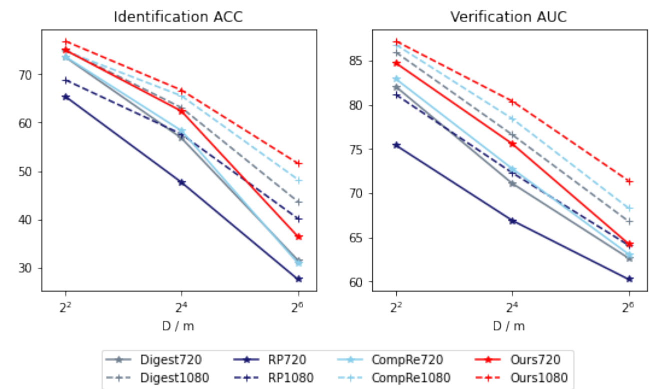

Fig. 4. Performance at different compression rate.

In our experiments on IMAGINE dataset, our method outperforms other compression methods at all three compression rates. Larger sized images are affected less from reduction on bit length.

4. CONCLUSION

In this paper, a two-stage PRNU fingerprint compression method is proposed. An auto-encoder based method is used to extract low-dimensional feature. Several experiments on IMAGINE show that using the compressed binary feature saves much storage space, speeds up the matching process and preserves the matching performance as well. Compared with other methods, ours achieves the best performance in the same settings. In general, our method can be employed to compress the extracted fingerprint feature, which can be used in many downstream tasks with preserved performance. However, our method takes longer time when compared to previous methods, and cropping is required to obtain images of fixed size that can be divided by block size, which increases complexity.

With regard to future work, an end-to-end neural network with integrated compression and binarization steps is an interesting topic, which may be more suitable for the task, and we believe better performance can be achieved by expansion of datasets.

5. REFERENCES

[1] Mustafa Al-Ani and Fouad Khelifi, "On the spn estimation in image forensics: a systematic empirical evaluation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1067–1081, 2016.

[2] Jessica Fridrich, Jan Lukas, and M Goljan, "Digital camera identification from sensor noise," *IEEE Transactions on Information Security and Forensics*, vol. 1, no. 2, pp. 205–214, 2006.

[3] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler, "Large scale test of sensor fingerprint camera identification," in *Media Forensics and Security*, Edward J. Delp III, Jana Dittmann, Nasir D. Memon, and Ping Wah Wong, Eds. International Society for Optics and Photonics, 2009, vol. 7254, p. 72540I, SPIE.

[4] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás, "Determining image origin and integrity using sensor noise," *Information Forensics and Security, IEEE Transactions on*, vol. 3, pp. 74 – 90, 04 2008.

[5] Samet Taspinar, Manoranjan Mohanty, and Nasir Memon, "Camera fingerprint extraction via spatial domain averaged frames," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3270–3282, 2020.

[6] Davide Cozzolino and Luisa Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.

[7] Mayank Tiwari and Bhupendra Gupta, "Image features dependant correlation-weighting function for efficient prnu based source camera identification," *Forensic science international*, vol. 285, pp. 111–120, 2018.

[8] Hui Zeng, Yongcai Wan, Kang Deng, and Anjie Peng, "Source camera identification with dual-tree complex wavelet transform," *IEEE Access*, vol. 8, pp. 18874–18883, 2020.

[9] Sara Mandelli, Davide Cozzolino, Paolo Bestagini, Luisa Verdoliva, and Stefano Tubaro, "Cnn-based fast source device identification," *IEEE Signal Processing Letters*, vol. 27, pp. 1285–1289, 2020.

[10] Giovanni Chierchia, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva, "A bayesian-mrf approach for prnu-based image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, 2014.

[11] Roberto Caldelli, Rudy Becarelli, and Irene Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, 2017.

[12] Jarosław Bernacki, "A survey on digital camera identification methods," *Forensic Science International: Digital Investigation*, vol. 34, pp. 300983, 2020.

[13] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler, "Managing a large database of camera fingerprints," in *Media forensics and security II*. SPIE, 2010, vol. 7541, pp. 75–86.

[14] Sevinç Bayram, Hüsrev Taha Sencar, and Nasir Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1404–1413, 2012.

[15] Ruizhe Li, Chang-Tsun Li, and Yu Guan, "Inference of a compact representation of sensor fingerprint for source camera identification," *Pattern Recognition*, vol. 74, pp. 556–567, 2018.

[16] Santosh S. Vempala, "The random projection method," in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2004.

[17] Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, and Enrico Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1472–1485, 2015.

[18] Diego Valsesia and Enrico Magli, "Binary adaptive embeddings from order statistics of random projections," *IEEE Signal Processing Letters*, vol. 24, no. 1, pp. 111–115, 2016.

[19] Luca Bondi, Paolo Bestagini, Fernando Perez-Gonzalez, and Stefano Tubaro, "Improving prnu compression through preprocessing, quantization, and coding," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 608–620, 2018.

[20] Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, and Enrico Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1439–1449, 2015.

[21] Xufeng Lin and Chang-Tsun Li, "Large-scale image clustering based on camera fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 793–808, 2016.

[22] Jarosław Bernacki, "Digital camera identification by fingerprint's compact representation," *Multimedia Tools Appl.*, vol. 81, no. 15, pp. 21641–21674, 2022.

[23] Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He, "Autoencoder and its various variants," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 415–419.

[24] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.

[25] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang, "Can we gain more from orthogonality regularizations in training deep cnns?," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2018, NIPS'18, p. 4266–4276, Curran Associates Inc.

[26] Andrew Brock, Theodore Lim, J.M. Ritchie, and Nick Weston, "Neural photo editing with introspective adversarial networks," in *International Conference on Learning Representations*, 2017.