IMAGE RETRIEVAL BASED ON COMPRESSED CAMERA SENSOR FINGERPRINTS

Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, Enrico Magli

Politecnico di Torino - DET, Italy
{name.surname}@polito.it

ABSTRACT

Image retrieval is the process of finding images from a large collection, satisfying a user-specified criterion. Content-based retrieval has been the traditional paradigm, in which one wishes to find images whose content is similar to a query. In this paper we explore a novel criterion for image search, based on forensic principles. We address the problem of retrieving all the photos in a collection that have been acquired by a specific device which is presented to the system as a query. This is an important forensic problem, whose solution could be very useful for detecting improper usage of pictures. We do not rely on metadata such as Exif headers because they can be unavailable, or easily manipulated, and in most cases cannot identify the specific device. We rely instead on a forensic tool called Photo Response Non-Uniformity (PRNU), which constitutes a reliable fingerprint of a camera sensor. We examine recent advances in compression of such fingerprints, which allow to address the previously unexplored image retrieval problem on large scales.

Index Terms— Random projections, PRNU, image forensics, image retrieval

1. INTRODUCTION

Millions of pictures are uploaded and shared over the Internet every day, creating a huge amount of data that calls for efficient solutions for its management. In this sense, very common tasks are the retrieval of pictures of interest and the classification of similar photos. Classical image retrieval techniques usually aim at finding pictures having similar content with respect to a query image [1]. However, forensic-oriented retrieval techniques may be of interest as well, for example if one wants to find pictures that have been acquired by a specific device.

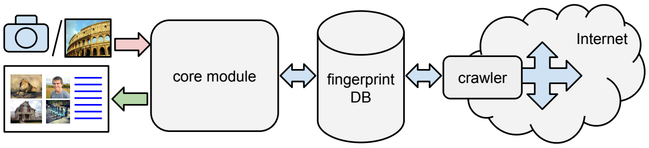

Fig. 1. High-level block diagram of the proposed system

If we imagine a search engine which, given as a query a specific device, returns all the webpages containing photos acquired by that device, this technology could be very useful for detecting improper usage of pictures. For example, professional photographers could use it to prevent improper diffusion of their photos, large websites could avoid being sued for redistributing unlicensed pictures, police investigators who have come across a digital camera or even just pictures linked to an unlawful act, e.g., child pornography, could look for other pictures taken by the same camera in either public databases (e.g., social networks) or large internal databases managed by the police.

When a picture is acquired by a digital sensor, slight imperfections in the manufacturing of single pixels produce a unique fingerprint, usually referred to as photo-response non-uniformity (PRNU) [2]. The PRNU can be considered as a noise-like, yet deterministic, pattern affecting every image taken by a sensor, and can be used to determine if a picture has been acquired by a given sensor, or if two or more pictures have been taken by the same camera. Several works demonstrate that the PRNU is a robust fingerprint, usually surviving processing like lossy compression and image resizing [3, 4].

The use of PRNU as a fingerprint for camera identification has so far focused on tasks involving a small number of cameras or photographs, as it is the case when it has to be used as evidence in a trial. Typically, in such scenarios one has to verify whether a picture, or a small set of pictures, has been taken by a specific camera and is concerned with having the best matching accuracy and low probability of false alarm. However, interesting scenarios involve the use of camera fingerprints on larger scales. One of them is using the PRNU for classifying a set of images according to the device that acquired them, which has received some attention in recent literature [5, 6, 7]. Another interesting problem, apparently overlooked in the literature, is retrieving all pictures in a large database that have been taken by a given camera.

In this paper, we present a problem of large-scale image retrieval based on PRNU fingerprints. The proposed system is a realistic example of the envisioned search engine, and can be used to demonstrate the the above technology is indeed feasible. The system is depicted in Fig. 1. We assume that a

This work is supported by the European Research Council under the European Communitys Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement n.279848.

large collection of photos is available, for example, it can be obtained by scanning portions of the web. From this collection, a large fingerprint database is automatically generated by extracting the PRNU pattern of each individual photo. A query is presented to the system in the form of a fingerprint of a camera, and the goal is to retrieve all the photos acquired by the same device.

Since the database could keep several millions of fingerprints, and PRNU patterns have the same size as the imaging sensor, which typically counts tens of millions of pixels, a technique for obtaining a compact representation of PRNU fingerprints is essential to deal with such a system. Notice that this image retrieval problem is significantly different from the problems addressed by the well-researched area of content-based image retrieval where the *content* of an image is the query and the user searches for images with similar content. Moreover, due to the specific properties of PRNU patterns, namely the fact that they are noise-like, traditional approaches, *e.g.*, based on feature descriptors [8] do not work for the presented problem.

In this paper, we propose to solve the above problem by using compressed fingerprints obtained via proper quantization of random projections. Recent works [9] have shown that random projections permit to obtain very compact representations with limited performance loss in terms of matching accuracy. Hence, this technique appears to be one of the best candidates for performing image retrieval based on camera fingerprints in very large scale scenarios.

The paper is organized as follows. Section 2 introduces the notation and provides background material on PRNU patterns. Section 3 describes the image retrieval problem and discusses in detail how an efficient implementation can be achieved with fingerprints compressed via random projections. Section 4 focuses on numerical experiments and section 5 draws some conclusions.

2. BACKGROUND

2.1. Notation

We denote (column-) vectors and matrices by lowercase and uppercase boldface characters, respectively. The ℓ-th element of column vector $\mathbf{v}$ is v_ℓ. The i-th column of the matrix $\mathbf{A}$ is $\mathbf{a}_i$. The notation $\mathbf{A} \cdot \mathbf{B}$ denotes the elementwise product between matrices $\mathbf{A}$ and $\mathbf{B}$, while $\mathbf{A}/\mathbf{B}$ denotes elementwise division. The notation $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the scalar product between vectors $\mathbf{a}$ and $\mathbf{b}$, and $\|\mathbf{a}\|_2 = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}$.

2.2. PRNU patterns

The PRNU of a digital imaging sensor constitutes a highly informative pattern that allows to uniquely identify a photographic device. The PRNU pattern is due to slight variations in the properties of individual pixels, which respond in a different manner to the incident light field, thus causing pixel-dependent gain variations. This pattern is deterministic, yet it has noise-like characteristics and it is unique of each sensor. It constitutes a robust fingerprint because it is stable in time, and survives processing like image compression, resizing, or many enhancement operations. It is thus possible to extract an estimate of the PRNU pattern of a particular sensor from one or more photos acquired. An acquired image $\mathbf{o}$ can be modeled as

$$\mathbf{o} = \mathbf{o}^{\mathsf{id}} + \mathbf{o}^{\mathsf{id}} \cdot \mathbf{k} + \mathbf{e} \,, \tag{1}$$

where $\mathbf{o}^{\mathsf{id}}$ is the ideal sensor output, $\mathbf{k}$ is the PRNU term and $\mathbf{e}$ collects other sources of noise. Assuming to be able to obtain through proper filtering a denoised version of $\mathbf{o}$, referred to as $\mathbf{o}^{\mathsf{dn}}$, then this can be used as an approximation of the ideal sensor output and subtracted from each side of (1) to obtain the so-called *noise residual*, which can be modeled as:

$$\mathbf{w} = \mathbf{o} - \mathbf{o}^{\mathsf{dn}} = \mathbf{o} \cdot \mathbf{k} + \tilde{\mathbf{q}} \,, \tag{2}$$

where $\tilde{\mathbf{q}}$ accounts for $\mathbf{e}$ and for the non-idealities of the model. Refer to [10] for more details on the model. It is thus seen that the noise residual is a modulated version of the PRNU where the modulating term is the current image, thus an estimate of the fingerprint $\mathbf{k}$ can be obtained by dividing the noise residual by the image. When C photos acquired by the same camera are available, the maximum likelihood estimate can be obtained as follows:

$$\hat{\mathbf{k}} = \sum_{\ell=1}^{C} \left(\mathbf{w}^{(\ell)} \cdot \mathbf{o}^{(\ell)} \right) \Big/ \sum_{\ell=1}^{C} (\mathbf{o}^{(\ell)})^2 \tag{3}$$

PRNU patterns have been successfully used in many forensic tasks such as determining whether a photo has been acquired by a specific camera. This is the most classic application of PRNU and it is often used in court trials since it has been declared admissible as evidence. Recently, variants of this problem have been studied in terms of classification or clustering problems, such as associating n photos to k cameras. As an example, a system is presented with n photos of unknown origin and it is told they come form k cameras, so that the task is determining the k clusters of photos according to the similarity of their PRNU patterns [5, 6, 7].

Few works recently started considering the problem of compression of PRNU fingerprints. In [11, 12], the authors introduced a so-called *fingerprint digest* obtained by keeping only a fixed number of the largest fingerprint values and their positions, which enables a fast search strategy with constant size fingerprints [13]. In [14], the authors proposed to represent sensor fingerprints in binary-quantized form: even though the size of binary fingerprints scales with sensor resolution, binarization can considerably speed-up the fingerprint matching process and considerably reduce the storage requirements. In [9] the authors proposed a compact representation of PRNU fingerprints based on a fixed number of random projections. Moreover, random projections can be binary quantized, leading to an extremely compact fingerprint representation. Thanks to the Johnson-Lindenstrauss (JL) lemma [15], results indicated that randomly-projected compressed fingerprints achieve identification performance close to the uncompressed fingerprints at a fraction of the storage space and considerably reduced computational cost for the matching operation. Moreover they are also a very flexible scheme in the sense that they allow to tune the number of projections to the desired tradeoff between speed and accuracy. They are thus the most promising technique to handle large scale scenarios. The details on how to use random projections to compress PRNU fingerprints are presented in the next section along with the novel contributions in this paper.

3. PROPOSED TECHNIQUE

3.1. Scenario

The system is given N photos of unknown origin, *e.g.*, they can be photos collected from web pages, and extracts a fingerprint estimate from each of them. Thus the system has to store N fingerprint estimates along with some metadata such as a URL for the actual photos. The system is queried with the fingerprint of a camera and has to return a list of photos from the database whose fingerprint estimate matches the one presented to the system. For simplicity, this paper considers perfectly synchronized images and fingerprints, thus having the original sensor resolution and being geometrically aligned so that no rotation or scale parameter has to be determined. The techniques presented in this paper can be extended to provide resilience to some geometric transformations. Some preliminary work in this direction can be found in [16].

The number N of fingerprint estimates to be stored can be very large, in the order of several millions for a real system, so compression techniques must be employed to have a manageable system.

3.2. Efficient implementation via random projections

The compression technique presented in [9] is based on computing a small number of random projections of a fingerprint and quantizing them. It was shown that binary-quantized random projections offer a very favorable tradeoff between detection accuracy and storage requirements. Essentially, this amounts to computing

$$\mathbf{y} = \mathrm{sign}(\mathbf{\Phi}\hat{\mathbf{k}}) \ , \tag{4}$$

where $\mathbf{\Phi}$ is a sensing matrix made of realizations of random variables of size $m \times n$, with $m \ll n$, thus mapping the original fingerprint $\hat{\mathbf{k}}$, composed of n pixels, to a lower-dimensional space through m measurements. A naive implementation of random projections poses significant problems of complexity since the most studied methods require a sensing matrix made of independent and identically distributed entries. This implies that mn random values must be either generated on-the-fly with a pseudorandom number generator or stored in memory. Either way, the large values of n (millions of pixels) and m (typically around 512000) make this approach impractical. Even worse, the computationally-expensive full matrix-vector product would be required to compute $\mathbf{y}$. The solution proposed in [9] is based on random partial circulant matrices. Those matrices are shown to perform very close to fully random matrices but allow fast implementations of the sensing operation because only n random values are to be generated and the operation $\mathbf{\Phi}\hat{\mathbf{k}}$ can be efficiently implemented via the FFT.

Hence, the system addressing the retrieval problem that we present in this paper, computes the quantized random projections for all the N collected fingerprints and stores them. Notice that the same matrix is used for all the fingerprints. The issue of handling sensors with multiple different resolutions is solved by using a submatrix of $\mathbf{\Phi}$ whose number of columns depends on the particular sensor resolution. Notice that, in this work, we are only concerned on verifying the performance of the retrieval based on fingerprint matching, so we do not exploit any side information such as sensor resolution which could indeed ease the task.

The query fingerprint is also compressed in the same way. The compressed query fingerprint is then compared with all the N compressed fingerprint stored in the database. Using binary-quantized random projections, the Hamming distance $d_H(\mathbf{a}, \mathbf{b}) = \frac{1}{m} \sum_{i=1}^{m} a_i \oplus b_i$, where $\oplus$ denotes the XOR operator, is used as dissimilarity metric. A match is declared when the Hamming distance is lower than a predefined threshold τ. The threshold can be set according to the desired probability of false alarm, *i.e.*, the probability that a non-matching fingerprint is incorrectly declared a match because its distance to the query randomly happens to be below the threshold.

Using the results in [9], we assume that the binary random projections $\mathbf{y}^{(i)}$ and $\mathbf{y}^{(j)}$ of two fingerprints of different devices are perfectly incoherent. This means that a single measurement (bit) is different with probability $\mathbb{P}\left(y_l^{(i)} \neq y_l^{(j)}\right) = \frac{1}{2}$. The measurements can be considered independent with good approximation, thus the Hamming distance follows a Binomial distribution, which is well approximated by a Gaussian with mean $\frac{1}{2}$ and variance $\frac{1}{4m}$. The false-alarm probability is readily obtained as a function of the threshold τ:

$$P_{FA} = \mathbb{P}\left(d_H(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) < \tau\right) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\tau - \frac{1}{2}}{\frac{1}{\sqrt{2m}}}\right) \tag{5}$$

Finally, notice that random projections significantly speed up the comparison operation as well as providing reduced storage requirements. This is due to two reasons. First, they provide an embedding into a lower dimensional subspace, thus the number of entries in a compressed fingerprint is significantly smaller than the number of pixels in the original (typical values are $m = 512000$ random projections and $n \approx 10^7$ pixels), so distances are more efficiently computed. Second, the Hamming distance is a very efficient operation because it just requires a XOR operation and a sum, thus requiring much fewer clock cycles than floating point multiplications needed to compute the correlation coefficient (or more complex metrics, like the Peak-to-Correlation Energy [10]) as it is done for uncompressed fingerprints.

3.3. A remark on Locality Sensitive Hashing

Locality Sensitive Hashing (LSH) is a technique used to solve the approximate nearest neighbor problem and it is indeed very successful at dealing with large-scale scenarios [17, 18]. This problem is concerned with finding the nearest neighbors, within a predefined distance, of a query point, among the N points in a database. LSH allows to create an efficient data structure to solve the problem, thus avoiding exhaustive search over the N points, and achieving sublinear complexity. In a nutshell, this is done by using locality-sensitive hash functions, *i.e.*, functions that return the same value with probability p_1 (ideally high) if two points are "close" to each other, or with probability p_2 (ideally low) if they are "far". The gap between the two probabilities is amplified by using multiple hash tables. Such system is able to achieve a $\mathcal{O}(N^\rho)$ retrieval complexity where $\rho < 1$ is the ratio between the radius of the ball enclosing the true neighbors and the distance beyond which the other points are supposed to lie. In principle, the retrieval problem presented in this paper well fits the class of problems solved by LSH. However, the peculiar characteristics of PRNU patterns make it impossible to use LSH efficiently. In fact, fingerprint estimates extracted from one or even multiple photos from the same device present very low correlation [14], thus making the factor ρ very close to 1. For example, the correlation coefficient of two matching uncompressed fingerprints is typically around 0.1. This implies a value of $\rho \approx 0.95$, so it is clear that LSH would not improve much over linear search. Moreover, the result is asymptotic and reaching it would require an impractically large number of hash tables and hash functions.

A final remark concerns the random projections used in this work and in [9]. Random projections are indeed used by LSH methods as a locality-sensitive function to create keys to hash tables. However, this work is only concerned with dimensionality reduction of the fingerprints to create a compact representation for storage and computational complexity reduction of the matching operations, without the objective to create a data structure to avoid exhaustive search.

4. NUMERICAL RESULTS

The performance of the compressed retrieval system is assessed using the publicly available database of photographs assembled by TU Dresden [19]. This database is well known in the forensic community as it is composed of both flatfield images, *i.e.*, photos of uniform subjects, that can be used to extract high quality fingerprint estimates, as well as natural images in outdoor and indoor environments. We selected 53 cameras having both flatfield images and natural images. A total of $N = 10965$ natural photos are available, and a fingerprint estimate is extracted from each of them. A query to the system is emulated by using fingerprint estimates extracted from the flatfield images. Each camera has a variable number of flatfield images, usually about 30 to 50, which are jointly used to extract a single high-quality estimate of the camera fingerprint. The Dresden image database is not very large but has been used in this paper for several reasons. First, it is well known in the literature, and being publicly available easily allows reproducible results. Moreover, we wanted to present a comparison with the ideal results of the uncompressed system, which quickly becomes unmanageable in terms of storage requirements and computational complexity as the size of the database grows.

We analyse three methods to solve the presented image retrieval problem. They are based on uncompressed fingerprints (single-precision floating point values), on the binarization method of [14] (1 bit per pixel), and the binary random projections of [9] (512000 binary-quantized random projections). Other methods exist, trying to address large scales and fast fingerprint matching, *e.g.*, some works based on fingerprint digests [13, 20, 21]. The simple fingerprint digest based on retaining the k entries of a fingerprint with largest magnitude was shown in [9] to have very similar performance to

Table 1. Size of the fingerprint database ($N = 10965$)

Uncompressed	Binary	Random projections
378.70 GB	11.83 GB	**669.25 MB**

Table 2. Complexity of query

	Uncompressed	Binary	Random projections
Time	154.8 sec	21.2 sec	**1.2 sec**
Cost	$\mathcal{O}(Nn)$ floating point multiplications	$\mathcal{O}(Nn)$ XOR	$\mathcal{O}(Nm)$ XOR

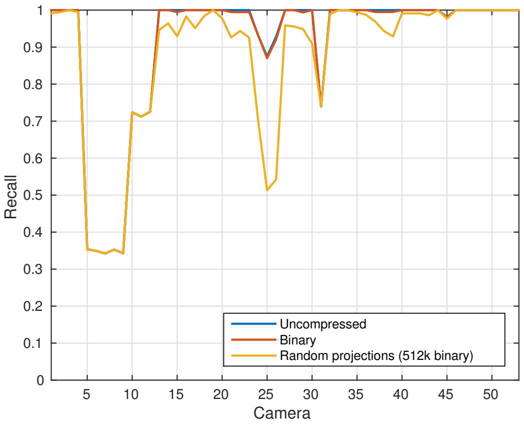

Fig. 2. Recall as function of query camera. Average recall: uncompressed = 0.912; binary = 0.911; random projections = 0.875.

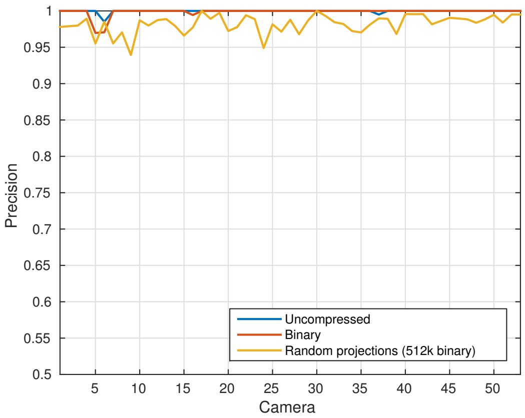

Fig. 3. Precision as function of query camera. Average precision: uncompressed = 0.999; binary = 0.998; random projections = 0.983.

real-valued random projections, but it was not competitive if compared to binary random projections. Some works studied data structures based on fingerprint digests that should accelerate the matching process when the fingerprints are highly correlated. However, in light of the aforementioned results, it is unclear how well they scale to million-entries datasets of fingerprints. A more exhaustive comparison with such methods will be the subject of future work.

Table 1 shows the amount of storage required for the fingerprint database using the three methods analysed in this section. In the following experiments, we compare each query fingerprint with all the fingerprints in the database. The uncompressed system uses as test metric the correlation coefficient between the uncompressed query fingerprint $\mathbf{d}^{(i)}$ and the database fingerprint $\mathbf{k}^{(j)}$.

$$c(i,j) = \frac{\langle \mathbf{d}^{(i)}, \mathbf{k}^{(j)} \rangle}{\|\mathbf{d}^{(i)}\| \|\mathbf{k}^{(j)}\|} \ , \ j = 1, \ldots, N$$

A match is declared when the correlation coefficient is above a predefined threshold, *i.e.*, $c(i,j) > \tau_{un}$. The binary fingerprints of [14] are compared using Hamming distance as dissimilarity metric. Since the image size may vary, the fingerprints are cropped to the smallest one. A match is declared when the Hamming distance is below a threshold τ_{bi}. Finally, the random projections technique uses compressed query and test fingerprints with $m = 512000$ binary-quantized random projections. A match is declared when the Hamming distance is below a threshold τ_{rp}.

The performance is evaluated in terms of recall and precision. The retrieved photos are those which have been declared a match according to the previously-defined test metrics. The recall is the number of retrieved photos acquired by the same camera of the query fingerprint, normalized by the total number of photos of that camera within the database. The precision is the number of retrieved photos acquired by the same camera of the query fingerprint, normalized by the total number of retrieved photos. Several ways of comparing the performance of the different techniques are possible. For example, a typical way would be setting the detection thresholds of the various methods in order to achieve the same false alarm probability, thus actually presetting the expected precision, as exemplified in Section 3.2. The experiments shown in Figs. 2 and 3 use a different method, based on using the threshold that gives the best F-score[1] for each of the methods. This choice allows to compare the methods at their best operating point in terms of precision-recall trade-off. Figs. 2 and 3 show the recall and precision values, respectively, as a function of the camera used as query. It can be noticed that a few cameras perform poorly, even in the uncompressed regime. This is a well known fact, as explained in [22], and it is due to non-unique artifacts introduced by onboard processing of the photos for some particular camera models under certain conditions (*e.g.* optical distortion correction when the optical zoom is used). We did not correct such artifacts, thus leaving the fingerprint extraction process blind to the actual camera models. The results show that the binary fingerprints obtain performance nearly indistinguishable from the uncompressed ones. However, their bit-size scales with the number of pixels in the camera, so each binary fingerprint requires about 1-2 MB of storage. Binary random projections use a fixed number of projections resulting in file sizes of about 64 kB per fingerprint. The results show that they are quite successful at achieving performance close to the uncompressed case, only suffering minor degradation in terms of precision and recall. Fig. 4 shows the precision-recall curve obtained using the average precision and recall over all the cameras and by sweeping the threshold values. Finally, Table 2 reports some figures about the time required to respond to a query for the various methods, as well as the computational complexity of such operation in big-O notation. Note that those times are based on our MATLAB implementation, which is not optimized for speed, and runs on a machine with 32 cores and 32GB of RAM. Moreover, they are estimated assuming that all the N fingerprints are loaded in RAM. This is indeed true for the random projections method and for the binarized fingerprints. However, since the N uncompressed fingerprints do not fit in the main memory, the query time of the uncompressed system is extrapolated from a subset of the N fingerprints. This highlights how the huge size of uncompressed fingerprints makes the retrieval problem highly impractical, requiring several machines in parallel or very long response times. On the contrary, quantized random projections allow very fast response times and modest memory requirements while having close-to-ideal performance.

5. CONCLUSIONS

In this paper, we presented an image retrieval problem where the goal is to retrieve photos acquired by a specific device in a large collection of photos. We showed how recent advances in compression techniques for PRNU patterns enable

[1] Calling precision P and recall R, the F-score is $F = 2\frac{PR}{P+R}$

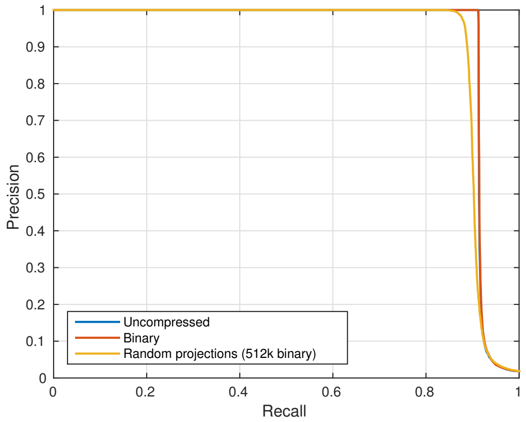

Fig. 4. Precision-Recall curve. Area under curve (AUC): uncompressed AUC = 0.9188; binary AUC = 0.9184; random projections AUC = 0.9061.

efficient solutions to this problem, achieving precision and recall performance close to the one of the uncompressed system at a fraction of storage and computational complexity. Future work should also address methods that enable resilience to geometrical transformations of the images and allow large-scale scenarios.

6. REFERENCES

[1] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Comput. Surv.*, vol. 40, no. 2, pp. 5:1–5:60, May 2008.

[2] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan, “Determining digital image origin using sensor imperfections,” in *Proc. SPIE Electronic Imaging, Image and Video Communication and Processing*, 2005, vol. 5685, pp. 249–260.

[3] J. Lukas, J. Fridrich, and M. Goljan, “Digital camera identification from sensor pattern noise,” *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 2, pp. 205–214, June 2006.

[4] Mo Chen, J. Fridrich, M. Goljan, and J. Lukas, “Determining image origin and integrity using sensor noise,” *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 1, pp. 74–90, March 2008.

[5] Chang-Tsun Li, “Unsupervised classification of digital images using enhanced sensor pattern noise,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS),*, May 2010, pp. 3429–3432.

[6] Bei bei Liu, Heung-Kyu Lee, Yongjian Hu, and Chang-Hee Choi, “On classification of source cameras: A graph based approach,” in *2010 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2010, pp. 1–5.

[7] I. Amerini, R. Caldelli, P. Crescenzi, A. Del Mastio, and A. Marino, “Blind image clustering based on the normalized cuts criterion for camera identification,” *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 831 – 843, 2014.

[8] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[9] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, “Compressed fingerprint matching and camera identification via random projections,” to appear in IEEE Transactions on Information Forsensics and Security, 2015.

[10] J. Fridrich, “Digital image forensics,” *Signal Processing Magazine, IEEE*, vol. 26, no. 2, pp. 26–37, 2009.

[11] Miroslav Goljan, Jessica Fridrich, and Tom Filler, “Managing a large database of camera fingerprints,” in *Proc. SPIE, Media Forensics and Security II*, 2010, vol. 7541, pp. 754108–754108–12.

[12] Yongjian Hu, Binghua Yu, and Chao Jian, “Source camera identification using large components of sensor pattern noise,” in *Computer Science and its Applications, 2009. CSA ’09. 2nd International Conference on*, Dec 2009, pp. 1–5.

[13] Yongjian Hu, Chang-Tsun Li, Zhimao Lai, and Shangfan Zhang, “Fast camera fingerprint search algorithm for source camera identification,” in *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, May 2012, pp. 1–5.

[14] S. Bayram, H.T. Sencar, and N. Memon, “Efficient sensor fingerprint matching through fingerprint binarization,” *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 4, pp. 1404–1413, 2012.

[15] William B Johnson and Joram Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert space,” *Contemporary Mathematics*, vol. 26, 1984.

[16] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, “Scale-robust compressive camera fingerprint matching with random projections,” to appear in the Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

[17] Piotr Indyk and Rajeev Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[18] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[19] Thomas Gloe and Rainer Böhme, “The Dresden Image Database for benchmarking digital image forensics,” *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

[20] Miroslav Goljan and Jessica Fridrich, “Sensor fingerprint digests for fast camera identification from geometrically distorted images,” in *Proc. SPIE, Media Watermarking, Security, and Forensics 2013*, 2013, vol. 8665, pp. 86650B–86650B–10.

[21] Yongjian Hu, Chang-Tsun Li, and Zhimao Lai, “Fast source camera identification using matching signs between query and reference fingerprints,” *Multimedia Tools and Applications*, pp. 1–24, 2014.

[22] Thomas Gloe, Stefan Pfennig, and Matthias Kirchner, “Unexpected Artefacts in PRNU-based Camera Identification: A ‘Dresden Image Database’ Case-study,” in *Proceedings of the on Multimedia and Security*. 2012, pp. 109–114, ACM.