Published as:

Bennabhaktula, G.S., Alegre, E., Karastoyanova, D. & Azzopardi, G. (2020) – *"Device-based image matching with similarity learning by convolutional neural networks that exploit the underlying camera sensor pattern noise"*, In Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, pp. 578-584.

Chapter 2

Device-based image matching for camera device identification from images

Abstract

One of the challenging problems in digital image forensics is the capability to identify images that are captured by the same camera device. This knowledge can help forensic experts gather intelligence about suspects by analyzing digital images. In this paper, we propose a two-part network to quantify the likelihood that a given pair of images has the same source camera, and we evaluated it on the benchmark Dresden data set containing 1851 images from 31 different cameras. To the best of our knowledge, we are the first ones addressing the challenge of device-based image matching. Though the proposed approach is not yet forensics-ready, our experiments show that this direction is worth pursuing, achieving at this moment 85 percent accuracy. This ongoing work is part of the EU-funded project 4NSEEK concerned with forensics against child sexual abuse.

2.1 Introduction

With the rapid adoption and consumption of digital content, there have been many instances of illicit material of children being circulated on the Internet, especially in the darknet. Today, Law Enforcement Agency (LEA) requires forensic tools that can help them to investigate more effectively and efficiently such digital content. The EU-funded 4NSEEK project [1], to which this work belongs, is aimed to develop a forensic tool by various partners in the industry and academia with the cooperation of police agencies in the European Union. The project is focused on fighting against child sexual abuse and the distribution of its contents across the internet. One desired functionality is device-based image matching, which is

[1] https://www.incibe.es/en/european-projects/4nseek

the determination of whether any two or more seized images were captured by the same camera device. Here we report the ongoing work in this direction.

Just as the bullet traces in a crime scene become a piece of evidence for a weapon, a digital image can become evidence for a camera. This is possible when we can extract fingerprints from images that (uniquely) characterize the source camera device. Extraction and identification of these fingerprints become more challenging when the photographs are subject to compression, post-processing, and computational photography, among others. Every processing step that alters the original RAW image, including the operations that are performed on the captured image within the camera, plays a role in altering the fingerprint. Together with the increasing use of image processing tools, the extraction of fingerprints becomes even more challenging.

The camera signature is embedded in the captured image in the form of noise and some artefacts. Our goal is to extract these fingerprints from given images and use them to determine whether the concerned images were captured by the same camera device. We would like to bring out a subtle difference between the terms camera model and camera device, with the former referring to the type of camera (e.g. Nikon D200) and the latter referring to a specific manufactured device (e.g. Nikon D200 - 1, where the last digit represents the unique identifier for the manufactured Nikon D200 devices). In this work, we address image matching by using signatures of the source camera devices.

More formally, the problem that we address in this work is the following: *given a pair of images, how likely are they both captured using the same camera device?* We restrict our analysis and discussions to the publicly available Dresden (Gloe and Böhme, 2010) image data set. We propose a Convolutional Neural Network (ConvNet) based architecture which is in line with the design of the ConvNet proposed by Mayer and Stamm (2018) for camera model identification.

The rest of the paper is organized as follows. We start by presenting an overview of the traditional and state-of-the-art approaches in Sec. 2.2. In Sec. 2.3, we describe the approach for feature extraction and classification of the proposed source camera identification. Experimental results along with the data set description are provided in Sec. 2.4. We provide a discussion of certain aspects of the proposed work in Sec. 2.5 and finally, we draw conclusions in Sec. 2.6.

2.2 Related work

The camera signature is embedded in the captured image in the form of noise and some artefacts. In Fig. 2.1 we illustrate a hierarchical representation of noise classification which we adopt from Lukas et al. (2006a). Even when the camera

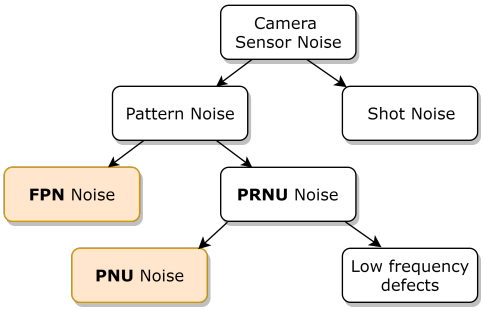

Figure 2.1: Topology of digital camera sensor noise. Note that only Fixed Pattern Noise (FPN) and Pixel Non-Uniformity noise (PNU), which are highlighted in yellow, contain the fingerprint that can be used to uniquely identify a sensor.

sensor is exposed to a uniformly lit scene the resulting image pixels are not uniform. This non-uniformity is caused due to shot noise and pattern noise. *Shot noise* is a temporal random noise and varies from frame to frame. This component of noise can be suppressed to a large extent by frame averaging. *Pattern noise* is defined as any noise component that survives frame averaging (Holst, 1998). This stability and uniqueness over time makes pattern noise a candidate for camera signature.

The two main components of pattern noise are Fixed Pattern Noise (FPN) and Photo Response Non-Uniform noise (PRNU). The FPN is an additive noise which is a consequence of dark currents (Holst, 1998). Dark currents are responsible for pixel-to-pixel differences when the sensor is not exposed to any light. Some modern digital cameras do offer long exposure noise reduction that automatically subtracts a dark frame from the captured image. This helps in removing the FPN artefacts from the captured image. This is, however, not a de-facto standard and is not implemented by all consumer camera manufacturers.

PRNU is further classified into Pixel Non-Uniformity noise (PNU) and noise caused by low-frequency defects. PNU noise is mainly caused due to imperfections and defects introduced into the sensor during the semiconductor wafer fabrication process. This in-homogeneity results in different sensitivities of pixels to light. The nature of PNU is such that even the sensors that are fabricated from the same wafer exhibit different PNU patterns. As mentioned by Lukas et al. (2006a), light refraction on dust particles, optical surfaces, and zoom settings also contribute to PRNU noise. These *low frequency* components are not characteristic of the sensor, hence they should be discarded when capturing the noise profile for a sensor from its image.

2.2.1 Traditional approaches

To the best of our knowledge, one of the earliest published works in camera detection was done by Geradts et al. (2001). The authors showed that every Charge Coupled Device (CCD) based sensor exhibits a few random pixels that are defective. These pixels can be identified under controlled temperatures. Repeated experiments showed that the location of such defective pixels always remains the same. The authors built a probabilistic model based on the location of defective pixels. The detection of such pixels is then left to visual inspection. Kharrazi et al. (2004) proposed 34 handcrafted features combined with an SVM classifier (Chang and Lin, 2011) to distinguish between images taken by Nikon E-2100, Sony DSC-P51, and Canon (S100, S110, S200) cameras. The authors extracted these features from both spatial and wavelet domains and carried out their experiments on a proprietary data set.

Kurosawa et al. (1999) were the first to consider FPN for source sensor identification. They established that this type of noise exhibits itself in images and is unique for each camera. The authors observed that the power of FPN is much less than the random noise. Hence, in order to suppress random noise and highlight FPN, they averaged 100 dark frames, which were captured by covering the camera lens. They performed experiments on nine different cameras, eight cameras of which exhibited FPN while the CCD-TRV90 Sony camera did not. Lukas et al. (2006a) have extended on this work by factoring in PRNU noise in addition to the FPN. For each camera under investigation, the authors generated a reference pattern noise, which serves as a unique identification fingerprint for the camera. The reference pattern is generated by averaging the noise obtained from multiple images using a denoising filter. The novelty of that approach is the generation of a camera signature without having access to the camera. Finally, a correlation was computed to establish the similarity between the query and reference patterns.

Li (2010) studied the noise patterns and observed that the scene details have stronger signal components while the true camera noise has weaker signals. Hence, they concluded, that stronger noise signal components in the residual image are less trustworthy. Based on this observation, an enhanced noise fingerprint was extracted by assigning less significant weights to stronger components of the noise signal.

A variety of methods have been proposed that account for Color Filter Array (CFA) demosaicing artefacts. Such methods identify the source camera of an image based on the traces left behind by the proprietary interpolation algorithm of each digital camera. Notable among these works include Bayram et al. (2005); Swaminathan et al. (2007), and a more recent one by Chen and Stamm (2015).

2.2.2 Approaches based on deep learning

In the last few years, deep learning based approaches have also been applied in the field of image forensics. Several ConvNet-based systems have been proposed to detect traces of image inpainting (Zhu et al., 2018), effects of image resizing and compression (Bayar and Stamm, 2017b), and median filtering detection (Chen et al., 2015), among other image forensic tasks.

Researchers have additionally proposed to apply ConvNets for the identification of the source camera of given images (Tuama et al., 2016; Bondi et al., 2016). Most of the deep learning algorithms follow an approach of extracting noise patterns by suppressing the scene content. Interestingly, the initial deep learning architectures for image denoising are inspired by the work in steganalysis (Qian et al., 2015). Bayar and Stamm (2016) used a high pass filter which could either be fixed or trainable as the first layer of a ConvNet and showed that this helps in scene suppression and helps the ConvNet to extract noise. Zhang et al. (2017) were the first ones to successfully perform residual learning using deep architectures. Residual learning is useful for camera sensor identification because the camera signature is often embedded in the residual images, which are obtained by subtracting the scene content from its image. The authors proposed a deep ConvNet model that was able to handle unknown levels of Additive White Gaussian Noise (AWGN). The ConvNet models were effective at denoising images with different types of noise, as opposed to traditional model-based designs (Kharrazi et al., 2004), which focused on detecting a specific type of noise.

The drawbacks of many of the proposed approaches are that they target specific types of forensic traces. For example, researchers have proposed methods that exclusively target CFA interpolation artefacts, chromatic aberration, assume a fixed level of Gaussian noise, and more. This is not an ideal assumption when developing real-world applications for forensic investigators. Here, we work with an open set of forensic traces.

The works which are very close to the ideas we propose are those by Cozzolino and Verdoliva (2019) and Mayer and Stamm (2018). Both approaches follow an open set of camera models. Many approaches that rely only on a closed set of camera models rely on prior knowledge from the source camera models. It looks almost impossible to use all existing camera models for training such models, and moreover, the scalability of such systems could be a challenge.

Cozzolino and Verdoliva (2019) designed a ConvNet which extracts a camera model fingerprint (as an image residue) known as the *noiseprint*. The authors use the ConvNet architecture proposed by Qian et al. (2015) and trained it in a Siamese configuration to highlight the camera-model artefacts. Their work primarily

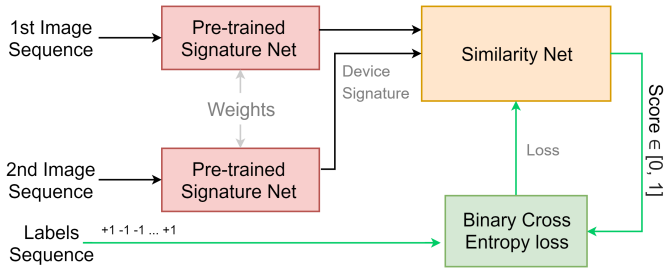

Figure 2.2: Proposed workflow.

focused on the extraction of noiseprint for camera models and on detecting image forgeries.

The ConvNet architecture that we adopt in our work is inspired by the work of Mayer and Stamm (2019). The authors have proposed a system called forensic similarity which determines if two image patches contain the same forensic traces or not. They proposed a two-part network. The first one is a feature extractor and the second part is a similarity network, which determines if two features come from the same source camera model. Patch-based systems do not account for the spatial locality. Therefore, instead of relying only on the patches, our proposed system takes the whole image for feature extraction. By considering the whole image the network has the possibility to learn the spatial locality in addition to the sensor pattern noise.

2.3 Proposed approach

The proposed method compares two input images and generates a score indicating the similarity between the source camera devices that took the concerned images. In Fig. 2.2 we depict the high-level workflow of the proposed method. The approach is divided into two phases. In the first phase, we train a ConvNet called henceforth as *signature network*, responsible for extracting the camera signature from an image. The second stage involves computing the similarity between two image signatures. The similarity function is formulated by training a neural network, which we call *similarity network*.

A two-phase learning approach gives us the ability to independently fine-tune signature extraction and similarity comparison. The training of the networks does not need the availability of ground truth noise residuals. It, therefore, allows us to have a more practical approach, as forensic investigators will not have access to the noise residuals for learning the camera signatures.

Table 2.1: The proposed ConvNet architecture of the signature network. It consists of 4 blocks of convolutional layers and 2 blocks of fully connected dense layers. The highlighted row indicates the layer at which we truncate the network and use the resulting 1024-element feature vector as the signature.

Layer #	Layers	Activation	Dimensions	Repeat
1	Convolutional (Conv) 2d Batchnorm Max pool	– tanh –	$96 \times 7 \times 7$ – 3×3	$\times 1$
2,3	Conv 2d Batchnorm Max pool	– tanh –	$64 \times 5 \times 5$ – 3×3	$\times 2$
4	Conv 2d Batchnorm Max pool	– tanh –	$128 \times 1 \times 1$ – 3×3	$\times 1$
5	Dense (signature)	tanh	1024	$\times 1$
6	Dense Dense	tanh softmax	200 # devices	$\times 1$

2.3.1 Learning phase I

The first phase in this approach begins with the training of a signature network, which is defined as follows.

Let the space of all RGB images be denoted by $\mathbb{I}$. The signature network is trained on a subset of images from $\mathbb{I}$. The trained network is then truncated at a feature extraction layer (Layer # 5, labelled Dense (signature) in Tab. 2.1), which we denote by f_{sig}. It is a feed-forward neural network function $f_{\text{sig}} : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{S}$, where $\mathbb{S}$ is a space of all signatures. We define the signature extraction operation, as follows:

$$S = f_{\text{sig}}(I) \tag{2.1}$$

$\forall\ I \in \mathbb{I}$, where $S \in \mathbb{S}$.

The signature network consists of four convolutional layers followed by two fully connected layers. A summary of all these layers is shown in Tab. 2.1. Note that the number of devices in the final fully connected layer represents the number of camera models present in the training set. The variable f_{sig} represents the trained network truncated at block 5 (see Tab. 2.1). This gives us a signature of 1024 elements in size.

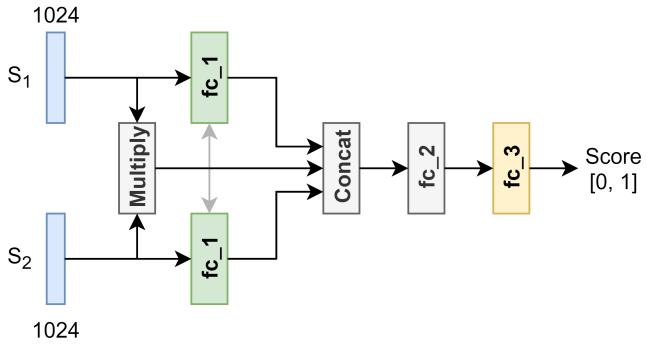

Figure 2.3: The proposed neural network architecture of the Similarity Network.

2.3.2 Learning phase II

The goal of the second phase is to map the signatures of pairs of images to a similarity score that gives an indication of whether the input pair comes from the same or different source. To this extent, we train a neural network in a Siamese fashion that determines the similarity between a pair of signatures extracted using the signature network. Let S_1 and S_2 be two signatures extracted from the signature network; $S_1 = f_{\text{sim}}(I_1)$ and $S_2 = f_{\text{sim}}(I_2)$. The labelled data for training the similarity network is then generated according to the following condition:

$$S_{\text{label}}(S_1, S_2) = \begin{cases} 1, & \text{If } I_1 \text{ and } I_2 \text{ come from the same source camera} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

The similarity network learns the mapping $f_{\text{sim}} : \mathbb{S} \times \mathbb{S} \rightarrow [0, 1]$, and its architecture is depicted in Fig. 2.3. The first layer is a fully connected dense layer fc_1 containing 2048 neurons with ReLU activation, which takes as input the signatures S_1 and S_2 of a given pair of images. Then, we combine the outputs from the first dense layer along with an element-wise multiplication of S_1 and S_2 into a single vector and feed it to fc_2, which is a dense fully connected layer with ReLU activations. This is finally connected to a single neuron with sigmoid activation. Once the similarity network is trained, we can use both networks together in a pipeline to determine the similarity for any given pair of input images.

$$\text{score} = f_{\text{sim}}(f_{\text{sig}}(I_1),\ f_{\text{sig}}(I_2)) \tag{2.3}$$

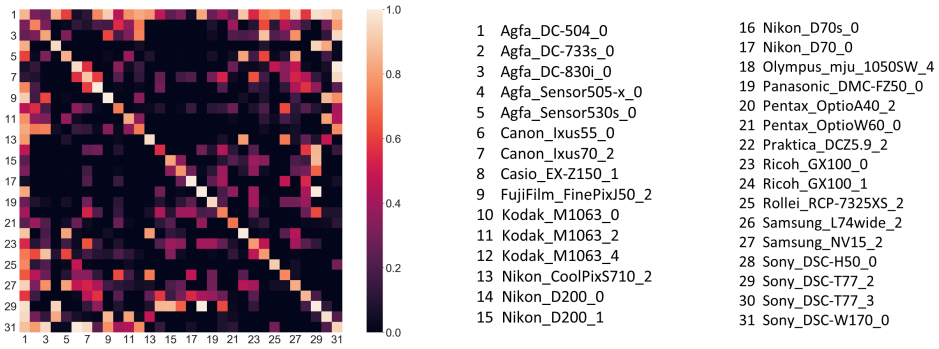

Figure 2.4: Similarity matrix for the 31 camera devices in the test set. A score closer to 1 indicates a high similarity between the images taken from the corresponding pairs of cameras. Similarity values along the diagonal correspond to the similarity between images taken from the same cameras. Ideally, the similarity matrix has ones along diagonal, and zeros elsewhere.

We experimentally determine a threshold η for the score given by the network. The pairs of images whose similarity score is above η are classified as similar, otherwise as different.

2.4 Preliminary experiments and results

2.4.1 Data set

We used the publicly available Dresden data set (Gloe and Böhme, 2010) in our experiments for source camera identification based on image matching. It consists of images from various indoor and outdoor scenes acquired under controlled conditions.

Many camera model identification approaches have been presented but due to a lack of a benchmark dataset, it is often hard to directly compare the performance of different methods. The Dresden data set was made available in 2010 and since then it has seen widespread use in image forensics that also goes beyond source camera identification. The Dresden data set comes with three subsets of data, one of which is called JPEG, which was intended for the study of model-specific JPEG compression algorithms. The JPEG set consists of 1851 images taken by 34 different camera devices that belong to 25 camera models. We discard the three camera devices (FujiFilm_FinePixJ50_0, Ricoh_GX100_3, Sony_DSC-T77_1) that contain only one image each and work with the remaining 31 devices. Though the image

content is limited to only two indoor scenes, it is of interest to understand the source camera device identification in the presence of JPEG compression artefacts. The other two subsets, which consist of dark frames and natural images, were not considered in this study.

2.4.2 Experiments

In a random stratified manner, 70% of the data was used for training and validation. The remaining 30% of data was considered as the test set.

In the first training phase, the signature network was trained for 15 epochs, with categorical cross entropy as the loss function and Stochastic Gradient Descent (SGD) as the optimizer. For the optimization task, a learning rate of 0.001, a momentum of 0.95, and a weight decay of 0.0005 were used. Convergence was reached after the 5^{th} epoch where the validation loss started to fluctuate while the training loss remained roughly the same, and we fixed the network with weights obtained at the end of the 5^{th} epoch. The training was done on an NVIDIA RTX 2070 GPU. All the extracted signatures were stored in a database, which provided easy access in the second half of the experiments.

In the second phase, where we train the similarity network, we generated labelled pairs of signatures according to Eq. (2.2). All the 1294 (70% of the full data set of 1851 images) training and validation images used for learning the signature network generated $\binom{1294}{2}$ pairs of labelled signatures data. The similarity network was trained in a Siamese fashion using binary cross entropy as the loss function along with SGD optimizer. The network was trained for 30 epochs, with a learning rate of 0.005 and a decay factor of 0.5 for every 3 epochs.

For the systematic evaluation of the trained network, a series of experiments were performed. A single experiment involves choosing a pair of camera devices and sampling 100 random pairs of images with replacement. Each pair consists of an image from each of the two concerned camera devices. The trained network was used to predict the similarity score for each of the pairs. The similarity score is converted to 1 (similar) or 0 (not similar) based on a threshold which we determined from the evaluation on the validation set. A threshold of 0.99 was set as it provided the maximum F1 score on the validation set. The resulting 100 scores were normalized by averaging them in order to get a value between 0 and 1 for the comparison of images coming from two camera devices.

For evaluation of the network, all possible pairs of the 31 camera devices were considered, which resulted in 31×31 experiments. Algorithm 2.1 was used to generate a similarity matrix of 31×31 elements, where each element is the normalized similarity score of the corresponding camera devices. Fig. 2.4 shows

the resulting similarity matrix on the test data. The overall test accuracy is 85%.

Algorithm 2.1 Similarity matrix computation.

procedure SIMILARITY MATRIX
  $\mathbf{C} \leftarrow \{C_1, C_2, ..., C_{\mathrm{N}}\}$ ▷ N cameras
  for $i \leftarrow 1 : \mathrm{N}$ and $j \leftarrow 1 : \mathrm{N}$ **do**
    `Randomly sample 100 image pairs`
    `from the subspace` $C_i \times C_j$

    `Predict the Source Similarity for`
    `the concerned 100 pairs of images`
    `using Eq. (2.3).`

    `Compute the accuracy`
  end for
  return accuracy for N × N experiments
end procedure

2.5 Discussion and future work

As can be seen from Fig. 2.4, in general, the model is able to detect images coming from the same camera devices. There are, however, some instances where the network gets confused with images coming from the same camera model. This can be seen with camera models (Ricoh_GX100_0, Ricoh_GX100_1), (Nikon_D70_0, Nikon_D70s_0). This could be because the same camera models are subject to the same manufacturing process. Thereby, resulting in similar imperfections or artefacts. We need to first investigate the noise differences between the same camera models, before trying to investigate the noise patterns together from all the devices. This approach might give us a better insight into the challenges between the same camera models.

It is also evident that the devices from the brands Agfa and Sony get confused with several other camera devices in our evaluation. We suspect this is due to the presence of a large number of images in the data set coming from Agfa (around 25 percent), which may have caused some bias in the learned networks. We will address this problem by investigating different approaches that deal with unbalanced training sets.

The proposed approach mimics the practical situation faced by forensic experts, where they only have a collection of images without knowing their actual source. Among others, investigators are interested to determine whether two or more images were taken by the same camera, irrespective of what camera it is. That

information can help them identify the offender or to compile stronger evidence. To the best of our knowledge, this is the first attempt that addresses the problem of device-based image matching.

2.6 Conclusions

From the results we achieved so far we conclude that the proposed approach is promising for matching images based on their underlying sensor pattern noise. We will continue our investigations and aim to improve the method until it is robust enough to be deployed as a forensic tool.