Forensic Source Camera Identification by Using Features in Machine Learning Approach

Amel Tuama Alhussainy

Collège Doctoral Languedoc-Roussillon

THÈSE

Pour obtenir le grade de
Docteur

Délivré par l'**Université de Montpellier**

Préparée au sein de l'école doctorale **I2S**
Et de l'unité de recherche **LIRMM**

Spécialité: **Informatique**

Présentée par **Amel Tuama ALHUSSAINY**

Forensic Source Camera Identification Using Features: a Machine Learning Approach

Soutenue le 1 Decembre 2016 devant le jury composé de

M.Jérôme Azé	Pr	Université de Montpellier	Président de Jury
M.Patrick Bas	DR-CNRS	Université de Lille	Rapporteur
M.Alessandro Piva	Associate Pr	Université de Florence (Italie)	Rapporteur
M.F. Pérez-Gonzàlez	Pr	Université de Vigo (Espagne)	Examinateur
M.Marc Chaumont	Mcf-HDR	Université de Nîmes	Directeur de thése
M.Frédéric Comby	Mcf	Université de Montpellier	Co-Encadrant
M.Alexis Joly	Mcf-HDR	Université de Montpellier	Invité

UNIVERSITE MONTPELLIER

Forensic Source Camera Identification by Using Features in Machine Learning Approach

Abstract:

Source camera identification has recently received a wide attention due to its important role in security and legal issue. The problem of establishing the origin of digital media obtained through an imaging device is important whenever digital content is presented and is used as evidence in the court. Source camera identification is the process of determining which camera device or model has been used to capture an image.

Our first contribution for digital camera model identification is based on the extraction of three sets of features in a machine learning scheme. These features are the co-occurrences matrix, some features related to CFA interpolation arrangement, and conditional probability statistics computed in the JPEG domain. These features give high order statistics which supplement and enhance the identification rate. The experiments prove the strength of our proposition since it achieves higher accuracy than the correlation-based method.

The second contribution is based on using the deep convolutional neural networks (CNNs). Unlike traditional methods, CNNs can automatically and simultaneously extract features and learn to classify during the learning process. A layer of preprocessing is added to the CNN model, and consists of a high pass filter which is applied to the input image. The obtained CNN gives very good performance for a very small learning complexity. Experimental comparison with a classical two steps machine learning approach shows that the proposed method can achieve significant detection performance. The well known object recognition CNN models, AlexNet and GoogleNet, are also examined.

Keywords: Camera Identification, PRNU, Co-occurrences, CFA interpolation, Deep Learning, Convolutional Neural Networks.

Identification d'appareils photos par apprentissage

Résumé :

L'identification d'appareils photos a récemment fait l'objet d'un grand intérêt en raison de son apport au niveau de la sécurité et dans le cadre juridique. Établir l'origine d'un média numérique obtenu par un appareil d'imagerie est important à chaque fois que le contenu numérique est présenté et utilisé comme preuve devant un tribunal. L'identification d'appareils photos consiste à déterminer la marque, le modèle, ou l'équipement qui a été utilisé pour prendre une image.

Notre première contribution pour l'identification du modèle d'appareil photo numérique est basée sur l'extraction de trois ensembles de caractéristiques puis l'utilisation d'un apprentissage automatique. Ces caractéristiques sont la matrice de co-occurrences, des corrélations inter-canaux mesurant la trace laissée par l'interpolation CFA, et les probabilités conditionnelles calculées dans le domaine JPEG. Ces caractéristiques donnent des statistiques d'ordre élevées qui complètent et améliorent le taux d'identification. La précision obtenue est supérieure à celle des méthodes de référence dans le domaine basées sur la corrélation.

Notre deuxième contribution est basée sur l'utilisation des CNNs. Contrairement aux méthodes traditionnelles, les CNNs apprennent simultanément les caractéristiques et la classification. Nous proposons d'ajouter une couche de pré-traitement (filtre passe-haut appliqué à l'image d'entrée) au CNN. Le CNN obtenu donne de bonne performance pour une faible complexité d'apprentissage. La méthode proposée donne des résultats équivalents à ceux obtenus par une approche en deux étapes (extraction de caractéristiques + SVM). Par ailleurs, nous avons examinés les CNNs : AlexNet et GoogleNet. GoogleNet donne les meilleurs taux d'identification pour une complexité d'apprentissage plus grande.

Mots clés : Identification de l'appareil source, PRNU, co-occurrences, Interpolation CFA, L'apprentissage en profondeur, Réseaux de neurones convolutif.

Dedication

My mother, your spirit is always with me pushing me forward.

Acknowledgement

I am grateful for Almighty Allah for giving me the strength and ability to fulfill this goal. My sincere appreciation and gratitude goes to my primary supervisor Dr. Marc Chaumont for his continuous and valuable advice and guidance, especially the constructive feedback on my thesis. My gratitude is also extended to my co-supervisor Dr. Frédéric Comby for his time, effort and helpful guidance.

Also, I would like to thank my examination committee: Professor Patrick Bas, Professor Alessandro Piva, Professor Frenando Pérez-Gonzàlez, Professor Jérôme Azé and Dr. Alexis Joly for their time and efforts.

I dedicate great thanks with love to my wonderful family, my husband Hasan Abdulrahman who taught me a great deal on both personal and professional levels. "I could not have achieved this without you by my side all the time". My lovely and beautiful daughter Dalya, thank you for your nonstop encouragement and understanding.

My friends, it is a blessing to have you in my life. Many thanks are also due to the administration staff in LIRMM laboratory, and all the members of the ICAR team for being such nice and cooperative.

Finally, I would like to thank The Iraqi Ministry of Higher Education and scientific research and the Iraqi Northern Technical University for funding my PhD thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Introduction

Today, multimedia (image, audio, video, etc) proceeds fast and spreads into all areas of human life. Frequent use of multimedia brings some new issues and challenges about its authenticity and reliability. Recent studies in multimedia forensics have begun to develop techniques to test the reliability and admissibility of multimedia.

In general, an evidence refers to information or objects that may be admitted into court for judges and juries to consider when hearing a case. An evidence can serve many roles in an investigation, such as to trace an illicit substance, identify remains or reconstruct a crime. The digital evidence is information stored or transmitted in binary form. It can be found on a computer hard drive, a mobile phone, a CD, and a flash card in a digital camera. For example, suspects e-mail or mobile phone files might contain critical evidence regarding their intent, their whereabouts at the time of a crime and their relationship with other suspects. For example, in 2005, a floppy disk led investigators to a serial killer who had eluded police capture since 1974 and claimed the lives of at least 10 victims [Jus15].

One of the multimedia elements is the digital image which is a very common evidence. An image (a photograph) is generally accepted as a proof of occurrence of the depicted event. As a way to represent a unique moment in space-time, digital images are often taken as silent witnesses in the court of law and are a crucial piece of crime evidence. Verifying a digital image integrity and authenticity is an important task in forensics especially considering that the images can be digitally modified by low-cost hardware and software tools that are widely available [TN08].

Section 1.2 of this chapter gives a definition and brief introduction about Digital Forensics. Image authentication and tamper detection is introduced in Section 1.3. A brief introduction to camera identification is given in Section 1.4. The objectives and contributions of this thesis will be presented in Section 1.5. The whole layout of this thesis is given in Section 1.6.

1.2 Digital Forensics

The first definition for digital forensics science has been formulated in 2001 during the first Digital Forensic Workshop [DFR01]. This definition was exactly: *"The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of* ***digital evidence*** *derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations".*

Digital forensics can simply be defined as the discipline that combines elements of law and computer science to collect and analyze data from computer systems, networks, wireless communications and storage devices in a way that is admissible as evidence in a court of law. In particular, digital forensics science emerged in the last decade in response to the escalation of crimes committed by the use of electronic devices as an instrument used to commit a crime or as a repository of evidences related to a crime [ACC+10].

The digital evidence is *any probative information stored or transmitted in digital form that a party to a court case may use at trial* [Cas04]. Digital forensics, as

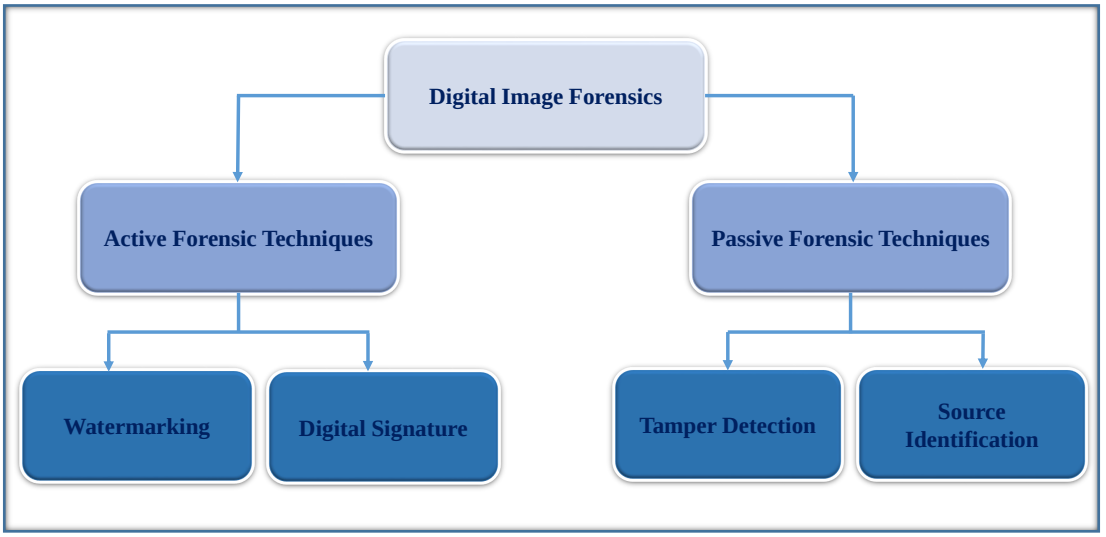


FIGURE 1.1: Hierarchy of digital image forensics.

illustrated in Figure 1.1, is divided into active and passive techniques. In the active forensic techniques, it is necessary to operate on the original document which has to be available from the beginning like in the watermarking or digital signature. While the passive forensic is a technique that can operate with no prior information about the content is available or no integrity protection mechanisms. It is straightforward to realize that this kind of investigation has to be founded on the thorough analysis of some intrinsic features that may be present inside the observed data [CH04].

Digital image forensics research aims at uncovering underlying facts about an image. It covers the answers to many questions such as:

- Can we trust an image?
- Is it original image or manipulated by some image processing tool?
- Was it generated by a digital camera, mobile phone, or a scanner?
- What is the brand and model of the source used to capture the image?

In digital image forensics, there are two main challenges. The first one is the source identification which makes possible to establish a link from the image to its source device, model, or brand. In tracing the history of an image, identifying the device used for its acquisition is of major interest. In a court of law, the origin of a particular image can represent a crucial evidence. The second challenge related to the detection of forgeries. In this case, it is required to establish if a certain image is authentic, or if it has been artificially manipulated in order to change its content [TN08].

1.3 Image Authentication and Tamper detection

With the rapid diffusion of electric imaging devices that enable the acquisition of visual data, almost everybody has today the possibility of recording, storing, and

sharing a large amount of digital images. At the same time, the large availability of image editing software tools makes extremely simple to alter the content of the images, or to create new ones, so that the possibility of tampering and modifying visual content is no more restricted to experts.

Tampering is any processing operation that is applied on a multimedia object after it has been created. Tampering can be divided to two types: innocent and malicious. Innocent tampering may modify the image quality in the time it doesn't change the contents of the image. This includes various operations such as contrast adjustment, brightness adjustment, up-sampling, downsampling, zooming, rotation etc. While malicious tampering aims at modifying the contents of the image and may includes operations such as cut-paste, copy-paste, region cloning and splicing [Ale13] as in the example illustrated in Figure 1.2.

FIGURE 1.2: Example of a tampered image: (a) The original picture of Ross Brawn receiving the Order of the British Empire from the queen Elizabeth II. (b) The tampered image depicting Jeffrey Wong Su En while receiving the award from the queen. The image was taken from [AWJL11].

Though existing digital forensic techniques are capable of detecting several standard digital media manipulations, they do not account for the possibility that may be applied to digital content. In reality, it is quite possible that a forger may be able to secretly develop anti-forensic operations and use them to create undetectable digital forgeries.

Anti-forensic or counter forensics operations designed to hide traces of manipulation and editing fingerprints resulted from forensic techniques. Research on counter-forensics is motivated by the need to assess and improve the reliability of forensic methods in situations where intelligent adversaries make efforts to induce a certain outcome of forensic analyses [BK13].

Furthermore, the study of anti-forensic operations can also lead to the identification of fingerprints left by anti-forensic operations and the development of techniques capable of detecting when an anti-forensic operation has been used to hide evidence forgery. It is clear that the authentication of multimedia signals poses a great challenge to information security researchers [Ale13].

1.4 Image Source Identification

As seen previously, source identification for digital content is one of the branches of digital image forensics. It aims at establishing a link between an image and its acquisition device by exploiting traces left by the different steps of the image acquisition process. Currently, the forensic community has put some efforts into the identification of images which may be generated by a digital camera, mobile phone, or even a scanner.

The authenticity of an image under investigation can be enforced by identifying its source. Source attribution techniques aim at looking for scratches left in an image by the source camera. These marks can be caused by factory defects, or the interaction between device components and the light.

In the source identification, the basic assumption is that digital contents are overlaid by artifacts added by the internal components of the acquisition device. Such artifacts are invisible to the human eye, but it can be analyzed to successfully contribute in the identification process. Source camera identification techniques achieve two major axes. The first one is searching for the properties of the camera

model, and the second is identifying the individual camera device [KG15]. The two axes will be explained in Chapter 2.

1.5 Thesis Objectives and Contributions

In this thesis, the subject of source device identification has been studied. In particular two different techniques will be presented in the following chapters. For each method, we describe all the conditions and details, and bring out the experiments and results that validate the methodology. The general aims and objectives of this thesis are as follows:

- Propose and analyze a technique for digital source camera model identification based on classical feature extraction and machine learning approach [TCC16a, TCC16b].
- Propose and implement the deep learning approach to enhance a CNN model for camera model identification.
- Investigate and demonstrate the state-of-the-art techniques related to source identification showing the limitations of each method.
- Compare our proposed methods performance with similar state-of-the-art techniques either in classical approach or in CNN approach.

1.6 Thesis Outline

This thesis is outlined as follows:

- Chapter 2 briefly highlight the recent state-of-the-art techniques for the camera identification forensics. Also we show a brief look to camera pipeline since it gives clues on where to find specific features in the acquisition process. Some traces are left that can be identified (or at least tried to be identified). The relationship with tamper detection and Anti-forensic is discussed.

- Chapter 3 discusses the global term for classical machine learning, then it goes deeper to illustrate the approach of deep learning and convolutional neural networks. Some details of Support Vector Machine are discussed.

- Chapter 4 describes the development of a method for digital camera model identification by extracting three sets of features in a machine learning scheme. These features are the co-occurrences matrix, some features related to CFA interpolation arrangement, and conditional probability statistics.

- Chapter 5 presents a new method of camera model identification using CNN approach. All the details of the proposed CNN architecture and System requirements are described. The experiments and comparisons with other models are demonstrated.

- Chapter 6 summarizes, concludes and discusses future work in camera identification.

- Chapter 8 is listing the international publications that support the work.

Chapter 2

Image Source Identification

Contents

2.1 Introduction

This chapter reviews digital forensic techniques for source camera identification. The tasks for digital multimedia forensics are grouped into six categories as follows [CFGL08]:

- Source Classification: classifies images according to their origin, scanner or camera device.
- Source Identification: searches for identifying a specific camera device, model, or make from a given image.
- Device Linking: links a device with a set of captured images.
- Processing History Recovery: retrieves the image processing steps applied to an image like type of compression method, or filtering.
- Integrity Verification and tamper detection.
- Anomaly Investigation: explaining anomalies found in images.

In our work, we focus on the source identification due to its necessity for legal and security reasons. Image source identification requires well understanding of the physical image formation pipeline. This pipeline is similar for almost all digital cameras, although much of the details are kept as proprietary information of each manufacturer.

This chapter will discuss in details these two groups. We will distinguish between the acquisition devices in the following section 2.2 followed by some details about the Exif Header of the image in the section 2.3. Methods of the first group will be discussed in Section 2.6 while the methods supported by machine learning will be discussed in Section 2.7.

2.2 Distinguishing between Acquisition Devices

The well known source of a digital image is the digital camera or cell phone device. Another kind of images that can constitute a digital evidence to be checked, in addition to those ones acquired with a photo camera or with a camcorder, might come from a scanning operation. This means that a printed document located in a flatbed scanner has been illuminated row by row by a sliding mono-dimensional sensor array to originate the digital data [KMC+07]. In this case, other elements, in addition to those already exist for cameras, can be considered during the forensic analysis process.

Computer generated graphics could be used to generate digital images since it touch many aspects of daily life. Computer imagery is found on television, in newspapers, and in all kinds of medical investigation and surgical procedures that has brought new challenges towards the originality of digital images. To locate the origin of the image whether it is a photographic or computer generated, image contour information can be extracted, or a correlation between CFA interpolation, or PRNU noise [PZ14]. Figure 2.1 illustrates the possible types of acquisition devices.

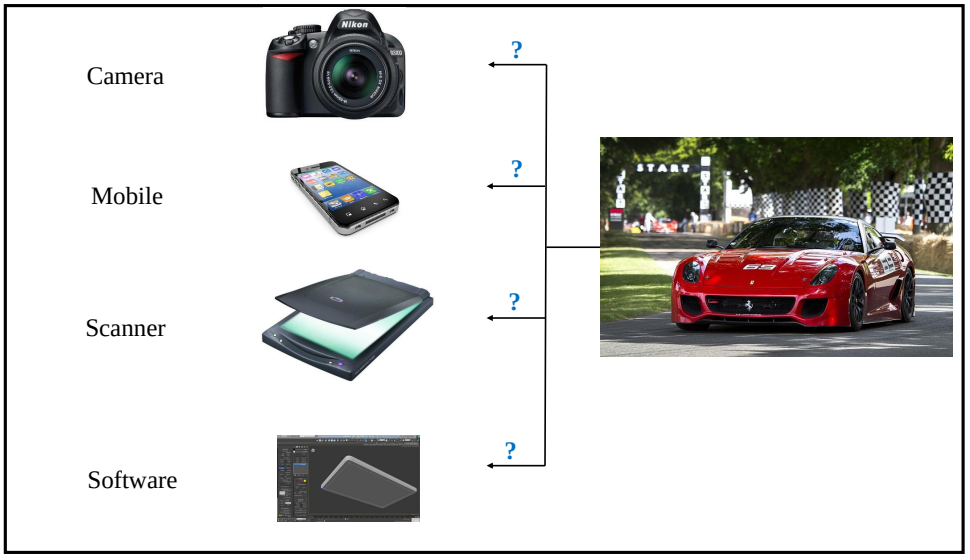


FIGURE 2.1: Types of acquisition devices.

In general, most of the techniques for camera identification do not work only for digital cameras but also for scanner and camcorder identification and also to distinguish between a photographic and a computer graphic image [TN08].

2.3 Exif Header of the image

Digital images, can be stored in a variety of formats, such as JPEG, GIF, PNG, TIFF. For example JPEG files contain a well-defined feature set that includes metadata, quantization tables for image compression and lossy compressed data. The metadata usually includes information about the camera type, resolution, focus settings, and other features [Coh07]. Besides when RAW format is used, the camera creates a header file which contains all of the camera settings, including sharpening level, contrast and saturation settings, colour temperature and white balancing.

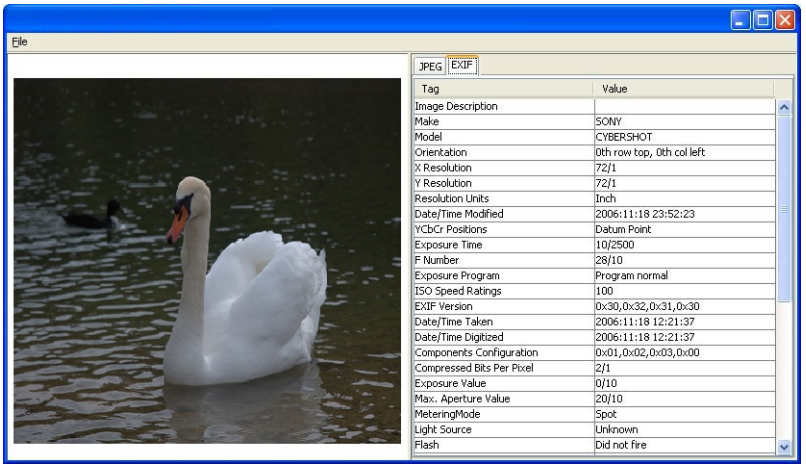


FIGURE 2.2: Some of the EXIF details

Although such metadata provide a significant amount of information it has some limitations since they can be edited, deleted and false information can be inserted about the camera type and settings. Normally, metadata or 'EXIF' header, refers to Exchangeable Image File Format, is considered the simplest way to identify an image source. It provides a standard representation of digital images as in Figure 2.2. Since the 'EXIF' headers can be easily modified or destroyed so we cannot rely on their information [TN08].

2.4 Camera pipeline and image formation

The general structure of a digital camera pipeline remains similar in all digital cameras. The exact processing detail in each stage varies from one manufacturer to the other, and even in different camera models manufactured by the same company. Figure 2.3 describes the basic structure for a digital camera pipeline. Digital Camera Pipeline consists of a lens system, optical filters, color filter array, imaging sensor, and a digital image processor.

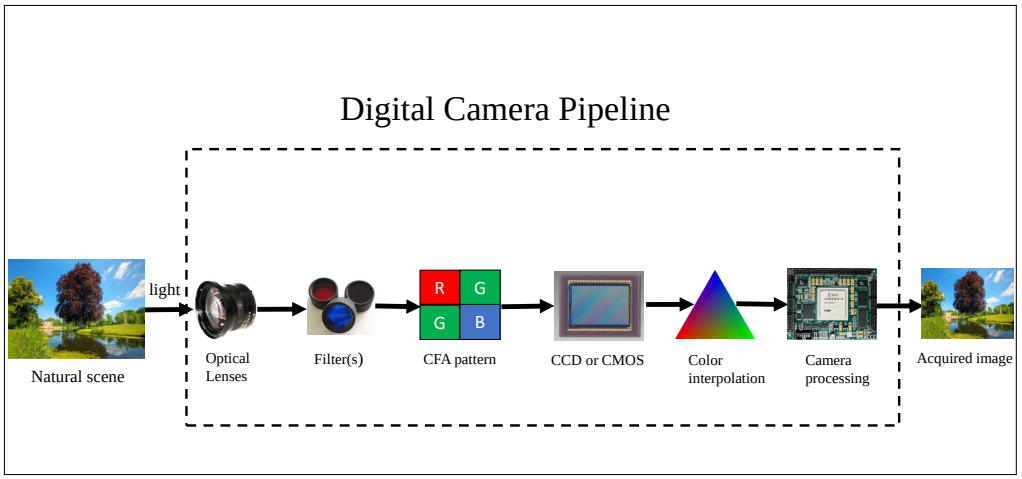


FIGURE 2.3: Image formation pipeline

- The lens system: It is essentially composed of a lens and the mechanisms to control exposure, focusing, and image stabilization to collect and control the light from the scene.

- Optical filters: After the light enters the camera through the lens, it goes through a combination of interposed optical filters that reduces undesired light components (e. g., infrared light).

- The imaging sensor: It is an array of rows and columns of light-sensing elements called photo-sites. In general there are two types of camera sensors deployed by digital cameras, the charge-coupled device (CCD) or complimentary metal-oxide semiconductor (CMOS). Each light sensing element of

sensor array integrates the incident light over the whole spectrum and obtains an electric signal representation of the scenery.

- CFA array: Since each imaging sensor element is essentially monochromatic, capturing color images requires separate sensors for each color component. However, due to cost considerations, in most digital cameras, only a single sensor is used along with a color filter array (CFA). The CFA arranges pixels in a pattern so that each element has a different spectral filter. Hence, each element only senses one band of wavelength, and the raw image collected from the imaging sensor is a mosaic of different colors and varying intensity values. The CFA patterns are most generally comprised of red-green-blue (RGB) color components.

- Demosaicing operation: As each sub-partition of pixels only provide information about a number of color component values, the missing color values for each pixel need to be obtained through demosaicing operation by interpolating three colors at each pixel location.

- Digital image processing: It is a series forms of image processing like white point correction, image sharpening, aperture correction, gamma correction and compression [KG15]. The colors are corrected, converting them from white balanced camera responses into a set of color primaries appropriate for the finished image. This is usually accomplished through multiplication with a color correction matrix. Edge enhancement or sharpening is applied on the image to reduce high spatial frequency content and improve the appearance of images. Once an image is fully processed, it is often compressed in one of the two compression algorithms; lossy and lossless. This step is important to reduce the amount of physical storage space required to represent the image data [AMB13].

2.5 State of the Art

In the literature of source identification, there are techniques that have investigated the presence of a specific CFA pattern in the image texture and other methods which have proposed to analyze the anomalies left by the device over the image such as scratches on the lenses, or defective pixels. In the other hand, big attention has paid to the defects of the sensor and dark current with what so called PRNU (Photo Response Non-Uniformity) which is one of the most interesting methods for forensic applications. While the other approaches use a set of data intrinsic features designed to classify camera models.

In source camera identification, it is necessary to distinguish between the levels of brand, model, and device. In order to well understand this hierarchical classification of cameras, in the first level of nomination process comes the name of the brand or manufacturer like the "Kodak" or "Nikon". The camera model comes in the second level such that models share most of the basic properties like the CFA pattern or lenses type and design. In the bottom of the classification hierarchy, we can see the individual digital cameras of the same model. Figure 2.4 shows these different levels with example from "Nikon" manufacturer.

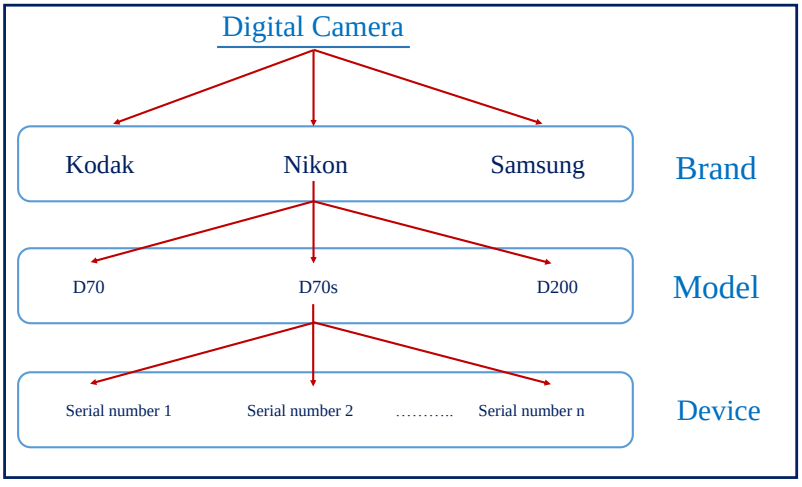


FIGURE 2.4: Camera levels, brand, model, Device.

Camera identification techniques are expected to achieve two major axes. The first extracts the model properties of the source, and the second is to identify the

individual source properties. Camera identification techniques based on sensor dust and sensor noise are only used for device identification while all the other methods are used for model identification [AWJL11].

We already classified the methods for identifying camera device into two main groups. The first group searches in camera identification through passing an statistical test between camera device and an image of unknown source. While the second group collects specific features from the image to deal with in a machine learning pool as shown in figure 2.5.

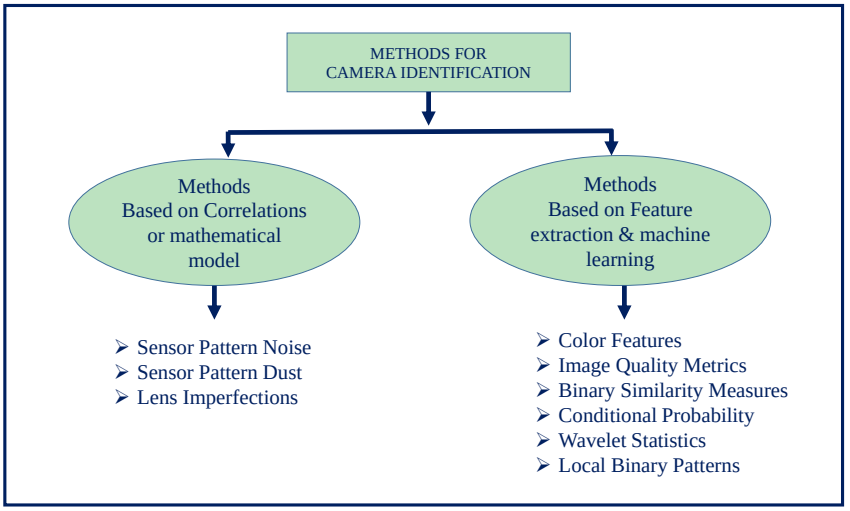


FIGURE 2.5: Camera Identification methods.

2.6 Methods based on a correlation test and mathematical model

This set of methods is based on producing a mathematical model in order to extract a relation between the image and the source. The image acquisition process involves many steps inside camera device which add artifacts to the image content. These artifacts can contribute in providing different features for the identification process. The techniques under consideration aim at analyzing those features in order to find a fingerprint for the device due to the sensor imperfections (dust and

noise), or lens aberrations. As a result the fingerprint of the camera is independent of the content of the analyzed data.

2.6.1 Sensor Pattern Noise

The imaging sensor consists of an array of rows and columns of light-sensing elements called photo-sites which are made of silicon. Each pixel integrates the incident light and converts photons into electrons by using analog to digital converter. There are many types of noise that comes from different factors including the imperfections in manufacturing process, silicone in-homogeneity, and thermal noise. The most significant one is the pattern noise. It is unique for each camera device which make it the only tool to identify an individual device. There are two main components of the pattern noise:

- The fixed pattern noise (FPN) is caused by dark currents when the sensor array is not exposed to light. It is an additive noise, so it is suppressed automatically by subtracting a dark frame from the captured image.
- The photo response non uniformity (PRNU) is the major source of noise. It is caused when pixels have different light sensitivities caused by the in-homogeneity of silicon wafers. PRNU is a high frequency multiplicative noise, generally stable over time and it is not affected by humidity and temperature.

The relation between the two types of pattern noise over an image I(x,y) is given in the equation 2.6 as follows:

$$I(x,y) = I_0(x,y) + \gamma I_0(x,y)K(x,y) + N(x,y) \tag{2.1}$$

where $I_0(x,y)$ is the noise-free image, γ is a multiplicative constant, $K(x,y)$ is the multiplicative noise or PRNU, and $N(x,y)$ is the additive noise or FPN.

2.6.1.1 Extracting PRNU

By correlating the noise extracted from a query image against the known reference pattern, or PRNU, of a given camera, we can determine whether that camera was used to originally capture the query image. The reference pattern of a camera is first extracted from a series of images taken from known camera device. The reference pattern is then used to detect whether the camera used to generate the reference pattern was used to capture an unknown source image. Generally, for an image I, the residual noise is extracted by subtracting the denoised version of the image from the image itself as follows:

$$N = I - F(I), \tag{2.2}$$

where $F(I)$ is the denoised image, and F is a denoising filter. A wavelet based denoising filter is used in most cases [Fri09]. In order to extract the fingerprint of a camera, multiple images are denoised and averaged. The averaging of multiple images reduces the random components and enhances the pattern noise. About 50 images are used to calculate the reference pattern K_d of a known camera device as in Equation 5.2.

$$K_d = \frac{\sum_{i=1}^{n}(N_i I_i)}{\sum_{i=1}^{n} I_i^2}. \tag{2.3}$$

A common approach to perform a comparison is to compute the Normalized Cross-Correlation which measures the similarity between the reference pattern K_d and the estimated noise N of an image under test which is of unknown source [Fri09]. Normalized Cross-Correlation is defined as:

$$\rho(N, K_d) = \frac{(N - \overline{N}).(K_d - \overline{K_d})}{\|N - \overline{N}\|.\|K_d - \overline{K_d}\|}. \tag{2.4}$$

Where $\overline{N}$ and $\overline{K_d}$ are the means of N and K_d, respectively.

2.6.1.2 Denoising Filter

Wavelet based denoising filter in the frequency domain is used in [LFG06] because it gave good results. By applying this particular denoising filter, the noise residual obtained contains the least amount of traces of the image content. The low frequency components of the PRNU signal are automatically suppressed when working with the noise residuals. Basically, this algorithm is composed of two steps. The first step estimates the local variance of the wavelet components, where the second step applies the Wiener filter on the wavelet coefficients [JM04]. An example of the results given by this filter is shown in figure 2.6. The denoising algorithm is as follows:

- Calculate the four level wavelet decomposition of the image using the Daubechies, 8-tap, *Separable Quadrate Mirror Filters* (QMF). The number of decomposition levels can be increased to improve accuracy or reduced to reduce processing time. At each level, the three high frequency sub-bands are horizontal H, vertical V, and diagonal D. For each wavelet sub-band, the local variance in a window of $(f \times f)$ of the neighborhood N is estimated by the formula in equation 2.5 as follows:

$$\hat{\sigma}_f^2(i,j) = max\left(0, \frac{1}{f^2} \sum_{(i,j)\in N} I^2(i,j) - \sigma_0^2\right), \tag{2.5}$$

  where $f \in \{3,5,7,9\}$, σ_0 is an initial integer constant value that we tuned manually such that $\sigma_0 \in 1, ..., 6$.

  The minimum value of the four variances will be taken as the final estimate.

$$\sigma^2(i,j) = min(\sigma_3^2, \sigma_5^2, \sigma_7^2, \sigma_9^2), \tag{2.6}$$

- The denoised wavelet coefficients are obtained using the Wiener filter mentioned in equation 2.7 for H, V, and D. Then, apply the inverse wavelet transformation on the denoised wavelet sub-bands to get the denoised component of the original image.

$$I_{clean}(i,j) = I(i,j)\frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j) + \sigma_0^2} \tag{2.7}$$

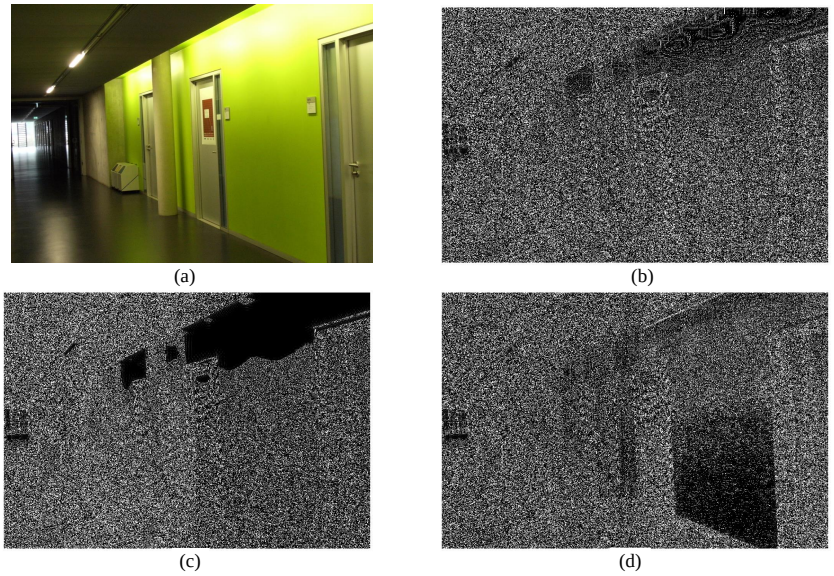


FIGURE 2.6: Denoising filter applied on a color image (a) the image , (b) denoised image of the red channel, (c) denoised image of the green channel, (d) denoised image of the blue channel.

2.6.2 Sensor Pattern Dust

This method is related to digital single lens reflex (DSLR) cameras which allow users to work with multiple interchangeable lenses. Once the lens is released, the dust particles are attracted to the camera sensor by electrostatic fields resulting a dust pattern which settles on the protective element at the surface of the sensor [DSM07]. The dust pattern can be seen as small specks, in the form of localized intensity degradations, all over the images produced by this camera device as shown in the Figure 2.7. Sensor dust spots can stay at the same position for very long times unless the sensor is cleaned. The random positions of dust spots create a unique pattern which can be used as a natural fingerprint of a DSLR camera. Dirik et al [DSM08] proposed the dust patterns as a useful fingerprint to identify an individual device. Dust spots in the image are detected based on a Gaussian intensity loss model and shape properties. The shape and darkness of the dust spots are determined by calculating the distance between the dust particle and

imaging sensor. The factors of camera focal length and aperture size are also used to determine the dust particles positions.

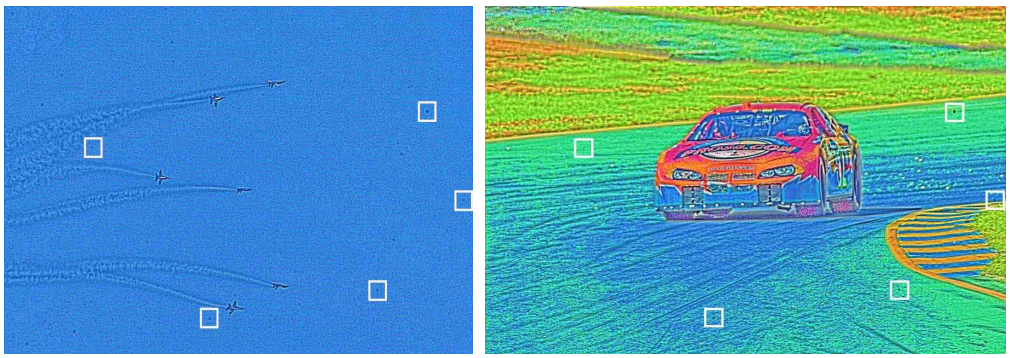


FIGURE 2.7: Dark spots in the white square are the sensor dust particles. The image was taken from [DSM08]

This method is not used with Compact cameras since they do not suffer from sensor dust problem. It is only used with DSLR cameras that need to be changed from time to another. In addition, recent devices come with built-in dust removal mechanisms.

2.6.3 Lens imperfections

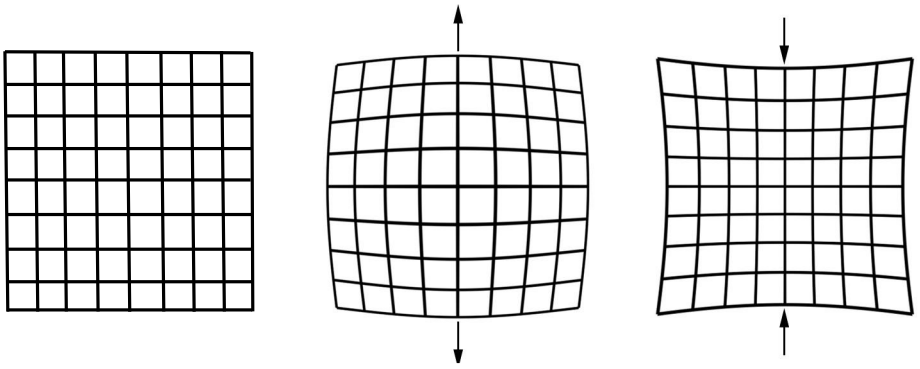
Each digital camera is equipped with a specific optical lenses to pass the scene light to the sensor. Most lenses introduce different kinds of lens aberrations such as spherical aberration, field curvature, lens radial distortion and chromatic distortion. Among the Lens aberrations, radial lens distortion is the most grave part [CLW06]

Due to the design process, most lenses introduce geometric distortion where straight lines in real world appear curved in the produced image. Figure 2.8 shows an example of geometrical lens distortion.

The radial distortion causes straight lines in the object space rendered as curved lines on camera sensor and it occurs when there is a change in transverse magnification M_t with increasing distance from the optical axis. The degree and the order of compensation of such a distortion vary from one manufacturer to another

a b

FIGURE 2.8: The distortion is clear in the first image.



a b c

FIGURE 2.9: Lens Radial distortion types (a)undistorted shape(b)barrel distortion(c)pincushion distortion

or even in different camera models by the same manufacturer. As a result, lenses from different cameras share unique fingerprints related to lens on the captured images. Lens radial distortion can be found in two types. The first one is called the barrel distortion, and it happens when M_t increases with r. The optical system suffers from pincushion distortion when M_t decreases with r. Example of the two types of distortion are shown in Figure 2.9.

The general formula of lens radial distortion can be written as in Equation 2.1:

$$r_u = r_d + k_1 r_d^3 + k_2 r_d^5 \tag{2.8}$$

where r_u and r_d are the undistorted radius and distorted radius respectively. The radius is the radial distance of a point (x, y) from the center of distortion, where $k1$ and $k2$ are the the distortion parameters. Choi et al [CLW06] proposed to

extract the distortion parameters following the Devernay's straight line method [FO01]. However this method fails if there are no straight lines in the image and also if two cameras of the same model are compared. Besides it is also possible to operate a software correction in order to correct the radial distortion on an image.

Another type of aberration is the chromatic aberration which is carried out to identify the source. Chromatic aberration is the phenomenon where light of different wave lengths fail to converge at the same position of the focal plane. There are two kind of chromatic aberration: longitudinal aberration that causes different wave lengths to focus at different distances from the lens, while lateral aberration is attributed at different positions on the sensor. In both cases, chromatic aberration leads to various forms of color imperfections in the image. Only lateral chromatic aberration is taken into consideration in the method described by Van et al [VEK07] for source identification. This method estimates the distorted parameters to compensate the distortion maximizing the mutual information among the color channels. Mayer et Stamm [MS16] proposed the lateral chromatic aberration for copy-paste forgery detection forensics. The authors proposed a statistical model of the error between local estimates of chromatic aberration displacement vectors and those predicted by a global model.

2.6.4 CFA pattern and Interpolation

Essentially, the sensor is monochromatic such that the capturing of a color image requires putting a color mask in front of the sensor. This is represented by the Color Filter Array (CFA) which it is a color mosaic that covers the imaging sensor.

The CFA permits only one color component of light to pass through it at each position before reaching the sensor. Each camera model uses one of several CFA patterns like those shown in figure 2.10. The most common array is the Bayer pattern which uses one red, one blue, and two green. RGBE pattern is used in some models of Sony cameras while CYYM pattern is used in some Kodak models.

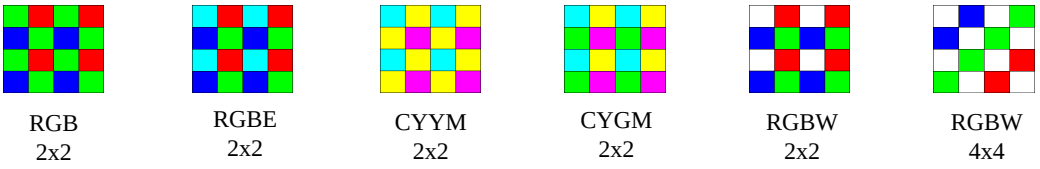


FIGURE 2.10: Color filter array patterns

As a result, the sensor records only one particular color value at each pixel location. The two missing color values at each pixel location must be estimated using a process known as demosaicing or color interpolation.

There are several algorithms for color interpolation such that each manufacturer employs a specific algorithm for a specific camera model. The source camera identification techniques are focused on finding the color filter array pattern and the color interpolation algorithm employed in internal processing blocks of a digital camera pipeline that acquired the image.

The approach proposed by Swaminathan et al. [SWL07] based on the fact that most of commercial cameras use RGB type of CFA with a periodicity of 2×2. Based on gradient features in a local neighborhood, the authors divided the image into three regions. Then they estimated the interpolation coefficients through singular value decomposition for each region and each color band separately. The sampled CFA pattern is re-interpolated and chose the pattern that minimizes the difference between the estimated final image and actual image produced by the camera.

Bayram et al. in [BSM06] proposed to use the periodicity arrangement in CFA pattern to find a periodic correlation. Expectation Maximization (EM) algorithm is applied to find the probability maps of the observed data to find a relation if exist between an image and a CFA interpolation algorithm. Two sets of features are extracted: the set of weighting coefficients of the image, and the peak locations and magnitudes in frequency spectrum. This method does not work in case of cameras of the same model, because they share the same CFA pattern and interpolation algorithm. Also, it does not work for compressed image, modified by gamma

correction, or smoothing techniques because these artifacts suppress and remove the spatial correlation between the pixels.

2.6.5 Camera Identification based Statistical Test

Another approach for camera identification is the statistical model-based methods. Thai et al. [TRC16] designed a statistical test within hypothesis testing framework for camera model identification from RAW images based on the heteroscedastic noise model. In this scenario, two hypotheses are proposed. The first one assumed that an image belongs to a camera A. While the second hypothesis assumed that the image belongs to a camera B. The parameters (a, b) are proposed to be exploited as camera fingerprint for camera model identification.

In an ideal context where all model parameters are perfectly known, the Likelihood Ratio Test (LRT) is presented and its statistical performances are theoretically established. In practice when the model parameters are unknown, two Generalized Likelihood Ratio Tests (GLRTs) are designed to deal with this difficulty such that they can meet a prescribed false alarm probability while ensuring a high detection performance.

The same statistical approach is used again for camera identification but this time it relies on the camera fingerprint extracted in the Discrete Cosine Transform (DCT) domain based on the state-of-the-art model of DCT coefficients [TRC15].

However, those prior works have the indisputable disadvantage to be unable of distinguishing different devices from the same camera model.

2.7 Methods based on feature extraction and machine learning

There are other approaches for camera model identification using a set of suitable digital data intrinsic features designed to classify a camera model. The feature

sets could be individual or merged. The promise of merging feature sets is that the resulting feature space provides a better representation of model specific image characteristics, and thus gives a higher classification accuracy than individual feature sets.

Kharrazi et al. [KSM04] proposed 34 features from three sets to perform camera model identification. The features are color features, Image Quality Metrics (IQM), and wavelet domain statistics. Image quality metrics (IQM) evaluated between an input image and its filtered version using a low-pass Gaussian filter, and integrated with color features (deviation from gray, inter-band correlation, gamma factor), and wavelet coefficient statistics. These features are used to construct multi-class classifiers with images coming from different cameras, but it is demonstrated that this approach does not work well with cameras with similar CCD and it requires images of the same content and resolution.

Celiktutan et al. [CSA08] used another group of selected features to distinguish among various brands of cell-phone cameras. Binary similarity measures (BSM), IQM features, and High-Order Wavelet Statistic (HOWS) features are used here to get 592 features.

Filler et al. [FFG08] introduced a camera model identification method using 28 features related to statistical moments and correlations of the linear pattern. Gloe et al. [Glo12] used Kharrazi's feature sets with extended color features to produce 82 features. Xu and Shi [XS12] used 354 Local Binary Patterns (or what so-called Ojala histograms) as features. Local binary patterns capture inter-pixel relations by thresholding a local neighborhood at the intensity value of the center pixel into a binary pattern.

Wahab et al. [AHL12] used the conditional probability as a single feature set to classify camera models. The authors considered DCT domain characteristics by exploiting empirical conditional probabilities of the relative order of magnitudes of three selected coefficients in the upper-left 4×4 low-frequency bands. The 72-dimensional feature space is composed from conditional probabilities over 8 different coefficient subsets.

Camera Iden. Method	Feature sets	No.of Features	No.of Models
Kharrazi et al. (2004) [KSM04]	Color features,IQM,Wavelet features	34	5
Celiktutan et al. (2008) [CSA08]	IQM,Wavelet features,BSM	592	16
Filler et al. (2008) [FFG08]	Statistical moments,Block covariance, Cross-correlation of CFA, Cross-correlation of linear pattern	28	17
Gloe et al. (2012) [Glo12]	Color features,IQM,Wavelet features	82	26
Xu and Shi (2012) [XS12]	Local Binary Patterns	354	18
Wahab et al. (2012) [AHL12]	Conditional Probability	72	4
Marra et al. (2015) [MPSV15]	Spam of Rich models	338	10

TABLE 2.1: A comparison between feature based camera identification methods.

Marra et al. [MPSV15] used of blind features extraction based on the analysis of image residuals. In this method, authors gathered 338 SPAM features (linear high pass filters computing derivatives of first to fourth order, called of type SPAM) from the rich models based on co-occurrences matrices of image residuals.

Table 2.1 shows the mentioned methods with their feature sets applied on a specific number of models.

2.8 Conclusion

This chapter covered a wide range of techniques used in forensic camera identification research field. The general structure of a digital camera pipeline and image formation is detailed to express the relation between camera pipeline and the method to identify a camera device. The main focus of the literature review was on camera identification techniques for digital images. We have classified techniques into two families: methods based on a statistical model and others based on feature machine learning model. In the next chapter, we will go on in the field of machine learning and more specific in CNN approach.

Chapter 3

Deep Convolutional Neural Networks

Contents

3.1 Introduction

Machine learning, as a sub-field of artificial intelligence, deals with intelligent systems that can modify their behavior in accordance with the input data [Vap95]. Intelligent systems must have the capability of deducing the function that best fits the input data, in order to learn from the data. In general, the approach of machine learning provides systems with the ability to learn from data by using repetition and experience, just like the learning process of humans. Depending on the information that is available for the learning process, the learning can be supervised or unsupervised. Supervised machine learning adopts the task of inferring a function from labeled and well identified training set. While unsupervised machine learning undertakes the inference process by using an unlabeled training set and seeks to deduce relationships by looking for similarities in the dataset [MTI15].

Features are used as essential key elements to complete the learning process. The feature is a quantitative measure that can be extracted from the digital media such that digital images. Preprocessing the image is done in order to put the feature set in a form accepted by the classifier.

Classification is defined as the process of identifying the class to which a previously unseen observation belongs, based on previous well trained dataset. The classification is giving the ability to distinguish between two or more classes by constructing a hyperplane among them. Any algorithm which performs mapping of input data to an assigned class is called a classifier. The training process makes use of a sample of N observations, the corresponding classes of which are certain. This sample of N observations is typically divided into two subsamples: the training and the test datasets. Firstly, the training dataset is used in the process of computing a classifier that is well-adapted to these data. Then the test dataset is used to assess the generalization capability of the previously computed classifier. K-nearest neighbors (KNN), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support vector machine (SVM) and multi-label classification support vector machine (libSVM) are commonly used Classifiers [Vap95].

This chapter discusses the global term for classical machine learning, then it goes deeper to illustrate the approach of deep machine learning as follows: Section 3.2 explains the details of Support Vector Machine SVM, with the problem of dimensionality and its available solutions. Section 3.3 deals with the approach of deep machine learning and convolutional neural networks (CNNs). We will list some of the popular CNN models in Section 3.4.

3.2 Classification by Support Vector Machine

In machine learning, a Support Vector Machine (SVM) is a supervised learning model. Its idea is to construct a hyperplane between two classes in a high dimensional space or infinite in some cases. SVM models are closely related to neural networks such that, a SVM model using a sigmoid kernel function is equivalent to a two-layer perceptron. The effectiveness of SVM depends on the selection of kernel function, and the kernel's parameters [HCL03]. The kernel function is the result of mapping two vector arguments into another feature space and then evaluating a standard dot product in this space. Kernel function could be Linear, Polynomial, Radial basis function (RBF), or Sigmoid function as shown in Table 3.1.

Kernel type	Formula
Linear	$K(X_i, X_j) = X_i.X_j$
Polynomial	$K(X_i, X_j) = (\gamma X_i.X_j + C)^d, \gamma > 0$
RBF	$K(X_i, X_j) = exp(-\gamma \lvert X_i - X_j \rvert^2), \gamma > 0$
Sigmoid	$K(X_i, X_j) = tanh(\gamma X_i.X_j + C)$

TABLE 3.1: Kernel function types of SVM classifier.

Using a kernel function provides a single point for the separation among classes. The radial basis function (RBF), which is commonly used, maps samples into a higher dimensional space that can handle the case when the relation between class labels and attributes is nonlinear.

3.2.1 The curse of dimensionality and Overfitting problem

In general, the performance of a classifier decreases when the dimensionality of the problem becomes too large. Projecting into high-dimensional spaces can be problematic due to the so-called curse of dimensionality. As the number of variables under consideration increases, the number of possible solutions also increases exponentially. The result is that the boundary between the classes is very specific to the examples in the training data set. The classifier has to handle the overfitting problem, so as it has to manage the curse of dimensionality [YON05]. The important question here is how to avoid or solve overfitting. Unfortunately, there is no fixed rule that defines how many feature should be used in a classification problem. In fact, this depends on the amount of training data available, the complexity of the decision boundaries, and the type of classifier used. In order to avoid overfitting caused by high dimensionality, the reduction of features would be a suitable solution. Since it is often intractable to train and test classifiers for all possible combinations of all features, several methods exist that try to find this optimum in different manners. These methods are called feature selection algorithms and often employ heuristics to locate the optimal number and combination of features such that Sequential floating forward selection method (SFFS), greedy methods, best-first methods.

A nother well known dimensionality reduction technique is Principal Component Analysis (PCA). PCA tries to find a linear subspace of lower dimensionality, such that the largest variance of the original data is kept. However, note that the largest variance of the data not necessarily represents the most discriminative information.

Cross validation can be used to detect and avoid overfitting during classifier training. Cross validation approaches split the original training data into one or more training subsets. During classifier training, one subset is used to test the accuracy and precision of the resulting classifier, while the others are used for parameter estimation. Several types of cross validation such as k-fold and leave-one-out cross-validation can be used if only a limited amount of training data is available. It is considered as a weak technique, because if the classification results on the training

subsets differ from the results on the testing subset, overfitting can't be prevented to occur.

3.3 Convolutional Neural Networks CNNs

Recently, Deep learning by using Convolutional neural networks CNNs have achieved wide interest in many fields. Deep learning frameworks are able to learn feature representations and perform classification automatically from original image. Convolutional Neural Networks (CNNs) have shown impressive performances in artificial intelligence tasks such as object recognition and natural language processing [BCV13].

Neural networks are nonlinear computational structures, modeled to behave like the humain brain, constructed of atomic components called *neurons*. Neurons can be defined using the McCulloch-Pitts Model [Zur92]. It consists of inputs, weights, a bias, an activation function, and the output. In general, the structure of CNN consists of layers which is composed of neurons. A neuron takes input values, does computations and passes results to next layer. The general structure of a CNN is illustrated in Figure 3.1 which also shows the similarities with traditional machine learning approach.

A Convolutional Neural Network is comprised of one or more convolutional layers and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units.

The output of this step will fed to the convolution layer to extract the feature map.

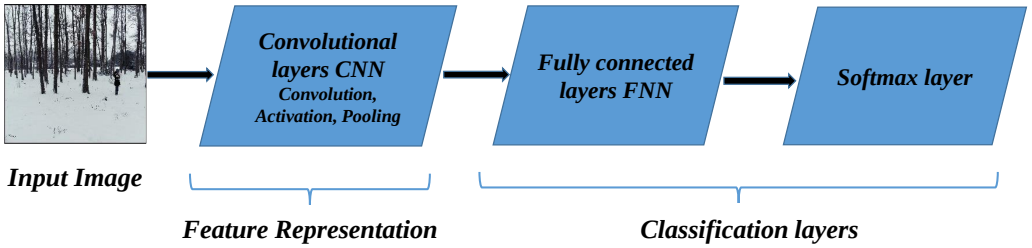


FIGURE 3.1: The Conventional neural networks concept

3.3.1 Convolutional layer

The convolutional layer is the core of the convolutional network. A conventional layer consists of three operations: convolution, the non-linearity activation function, and pooling. The result of a convolutional layer is called feature map which can be considered a particular feature representation of the input image. The input to a convolutional layer is an image of $m \times m \times r$ where m is the height and width of the image and r is the number of channels. The convolutional layer will have k filters or kernels of size $nxnxq$ where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters convoluted with the image to produce k feature maps of size $m-n+1$. Each map is then sub-sampled typically with mean or max pooling over pxp contiguous regions where p ranges between 2 for small images and is usually not more than 5 for larger inputs. Either before or after the pooling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The convolution can be formulated as follows [PPIC16]:

$$a_j^l = \sum_{i=1}^{n} a_i^{l-1} * w_{ij}^{l-1} + b_j^l, \tag{3.1}$$

where $*$ denotes convolution, a_j^l is the j-th output map in layer l, w_{ij}^{l-1} is convolutional kernel connecting the i-th output map in layer $l-1$ and the j-th output map in layer l, b_j^l is the training bias parameter for the j-th output map in layer l.

3.3.2 Activation function

The *activation function* is applied to each value of the filtered image. There are several types of the activation function such as, an absolute function $f(x) = |x|$, a sine function $f(x) = sinus(x)$, or Rectified Linear Units (ReLU) function $f(x) = max(0, x)$.

Rectified Linear Units (ReLUs) is the non-linearity activation function which is applied to the output of every convolutional layer. ReLUs is considered as the standard way to model a neuron's output and it can lead to fast convergence in the performance of large models trained on large data sets [PPIC16].

3.3.3 Pooling

The next important step in the convolution process is the pooling. A pooling layer is commonly inserted between two successive convolutional layers. Its function is to reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. It is considered as a form of non-linear down-sampling. Max-pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value. Max pooling propagates the average and the maximum value within the local region to the next layer. The loss of spatial information is translated to an increasing number of higher level feature representations. Max-pooling is useful in vision for two reasons:

It provides a form of translation invariance. Imagine cascading a max-pooling layer with a convolutional layer. There are 8 directions in which one can translate the input image by a single pixel. If max-pooling is done over a 2x2 region, 3 out of these 8 possible configurations will produce exactly the same output at the convolutional layer. For max-pooling over a 3x3 window, this jumps to 5/8.

Since it provides additional robustness to position, max-pooling is a "smart" way of reducing the dimensionality of intermediate representations. The last process

done by the layer is a normalization of the feature maps applied on each value. The normalization is done across the maps, which is useful when using unbounded activation functions such as ReLU [PPIC16].

3.3.4 Classification by Fully connected layers

In general, the classification layer consists of the fully connected layers. Fully connected layers mean that every neuron in the network is connected to every neuron in adjacent layers. When the learned features pass through the fully connected layers, they will be fed to the top layer of the CNNs, where a softmax activation function is used for classification. The back propagation algorithm is used to train the CNN. The weights and the bias can be modified in the convolutional and fully connected layers due to the error propagation process. In this way, the classification result can be fed back to guide the feature extraction automatically and the learning mechanism can be established.

3.3.5 Learning process and Back-propagation algorithm

The back-propagation algorithm consists of forward and backward passes. First, the model calls forward pass to yield the output and loss, then calls the backward pass to generate the gradient of the model, and then incorporates the gradient into a weight update that minimizes the loss.

The forward pass computes the output given the input for inference by composing the computation of each layer to compute the function represented by the model. The general optimization problem of the model depends on the loss minimization. Given the dataset S, the optimization objective is the average loss over all $|S|$ data instances throughout the dataset:

$$L(W) = \frac{1}{|S|}\Sigma_i^{|S|} Lw(X^{(i)}) + \lambda r(W). \tag{3.2}$$

Where $Lw(X^{(i)})$ is the loss on data instance $X^{(i)}$, and $r(W)$ is a regularization term with weight λ [JSD+14].

The backward pass computes the gradient given the loss for learning where the model reverse-composes the gradient of each layer to compute the gradient of the whole model by automatic differentiation. Stochastic gradient descent (SGD) is a radical simplification algorithm that provides a good learning when the training set is large. The SGD updates the weights by a linear combination of the negative gradient $\nabla L(W)$ and the previous weight update V_t.

The following formulas are used to compute the update value V_{t+1} and the updated weights W_{t+1} [Bot12]:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t). \tag{3.3}$$

$$W_{t+1} = W_t + V_{t+1}. \tag{3.4}$$

where α is the learning rate, μ is the momentum term, Vt is the previous weight update, and Wt is the current weight. The learning rate α has to be initialized to a value around 0.01 and μ to 0.9, and then they might be tuned for best results [KSH12].

3.3.6 Drop-out technique

The CNN architecture has thousands of parameters which may arise overfitting problem. Drop out technique is used for reducing overfitting. It consists of setting the output of each hidden neuron with probability 0.5 to zero. The neurons which are dropped out in this way do not contribute to the forward pass and do not participate in backpropagation. This technique reduces complexity, since a neuron cannot rely on the presence of particular other neurons. It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

3.4 Examples of CNNs

There are several CNN architectures designed to perform in different scientific fields. The most common are:

- LeNet was the first application of Convolutional Networks developed by Yann LeCun in 1990's [LBBH98] which was used to read zip codes, digits, etc.
- AlexNet is the first CNNs in Computer Vision, developed by Alex Krizhevsky et al. [KSH12]. The network design has a very similar architecture to LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other. Figure 3.2 illustrates the structure of AlexNet CNN model.

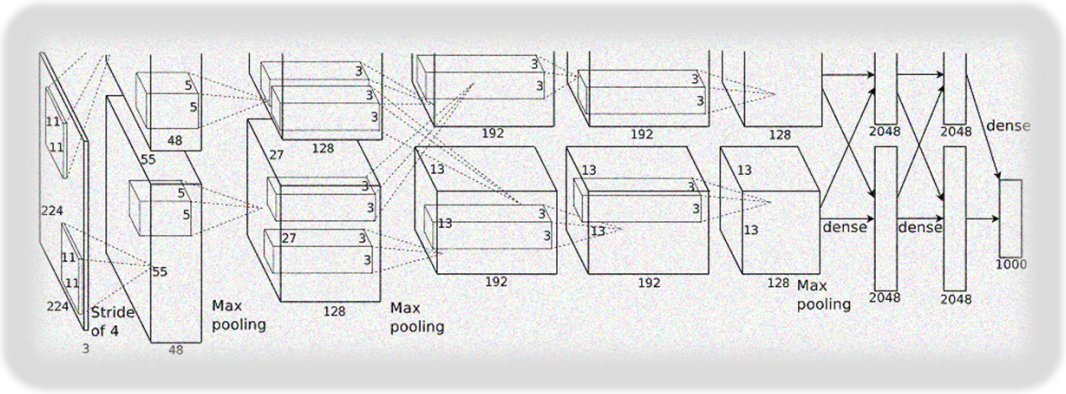


FIGURE 3.2: AlexNet CNN model with the use of 2 GPUs. Image is extracted from [KSH12]

- ZF Net was the winner in the ILSVRC 2013 challenge, designed by Matthew Zeiler and Rob Fergus [ZF14]. It was an improvement on AlexNet by tweaking the architecture hyper parameters, in particular by expanding the size of the middle convolutional layers and making the stride and filter size on the first layer smaller.
- GoogLeNet was the winner of the ILSVRC 2014 challenge designed by Szegedy et al [SLJ+15]. Its main contribution was the development of an Inception Module that dramatically reduced the number of parameters in the network

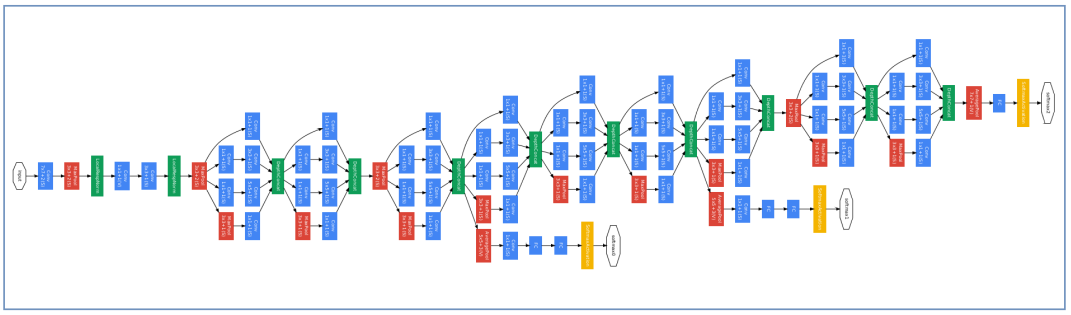


FIGURE 3.3: The layout of GoogleNet. Image is extracted from [SLJ+15]

(4M, compared to AlexNet with 60M). Additionally, this paper uses Average Pooling instead of Fully Connected layers at the top of the ConvNet, eliminating a large amount of parameters that do not seem to matter much. Figure 3.3 illustrates the structure of GoogleNet CNN model (27 convolutional layers).

- VGGNet: It was the runner-up in ILSVRC 2014 developed by Karen Simonyan and Andrew Zisserman [SZ14]. Its main contribution was in showing that the depth of the network is a critical component for good performance. Their final best network contains 16 CONV/FC layers and, appealingly, features an extremely homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end. A downside of the VGGNet is that it is more expensive to evaluate and uses a lot more memory and parameters (140M). Most of these parameters are in the first fully connected layer, and it was since found that these FC layers can be removed with no performance downgrade, significantly reducing the number of necessary parameters.

- ResNet: Residual Network developed by Kaiming He et al [HZRS15] which was the winner of ILSVRC 2015. It features special skip connections and a heavy use of batch normalization. The architecture is also missing fully connected layers at the end of the network.

3.5 Conclusion

Convolutional Neural networks (CNNs) are an extended version of neural networks. In this chapter, a brief look to SVM classification is included. We present an overview about CNNs and how they are implemented. CNNs are able to extract and learn features from the input data directly through hierarchical convolutional layers. The output of these hierarchical feature extractors is connected to a fully-connected neural network that performs a classification task.

In the field of digital forensics Bayar et Stamm [BS16] proposed a deep learning approach to detect image manipulation, while Chen et al. [CKLW15] introduced the convolutional neural networks in median filtering forensics. We can see that CNN approach has not been used for camera identification. CNNs can act a good tool for camera identification process. In Chapter 5, we used the approach of CNNs to perform a new camera identification method.

Chapter 4

Source Camera Model Identification Using Features from Polluted Noise

Contents

4.1 Introduction

Source camera identification is one of the major interests in image forensics. It is the process of deciding which camera has been used to capture a particular image. The methods for camera identification can be categorized into two main families. The first family is based on producing a fingerprint, for example a PRNU. The correlation between a given image and the fingerprint of a specified camera can then be computed. The second family regroups the methods based on machine learning and feature vector extraction. Here, the model is built by the classification algorithm knowing the features. In order to identify a camera, the classifier evaluate the proximity (distance) between a previously learned model, and the feature vector of the image to test. Our approach is a mix of the two families since we use a residual, we referred to by *polluted PRNU*, in a machine learning approach. We developed a method for digital camera model identification by extracting three sets of features in a machine learning scheme. These features are the co-occurrences matrix, some features related to CFA interpolation, and conditional probability computed in the DCT domain. These features give high order statistics which supplement and enhance the identification rate.

This chapter is organized as follows: Section 4.2 explains the classical approach to compute a PRNU. Section 4.3 presents all the details of the proposed machine learning approach. Subsections describe how to extract the POL-PRNU residual, the feature set 1 (spatial co-occurrences), 2 (CFA interpolation traces), and 3 (frequential Conditional probability). In section 4.4, we describe the experiments, the results, and the database used for experiments. Conclusion comes in Section 4.5.

4.2 Correlation method for camera identification (PRNU)

Sensor pattern noise has drawn much attention due to its feasibility in identifying camera models of the same brand, and individual devices of the same model. The PRNU is unique to each sensor and is stable over time. By correlating the noise extracted from a query image against the known reference pattern, or PRNU, of a given camera, we can determine whether that camera was used to originally capture the query image. The reference pattern of a camera is first extracted from a series of images taken from known camera device. The reference pattern is then used to detect whether the camera is used to generate an unknown source image or not.

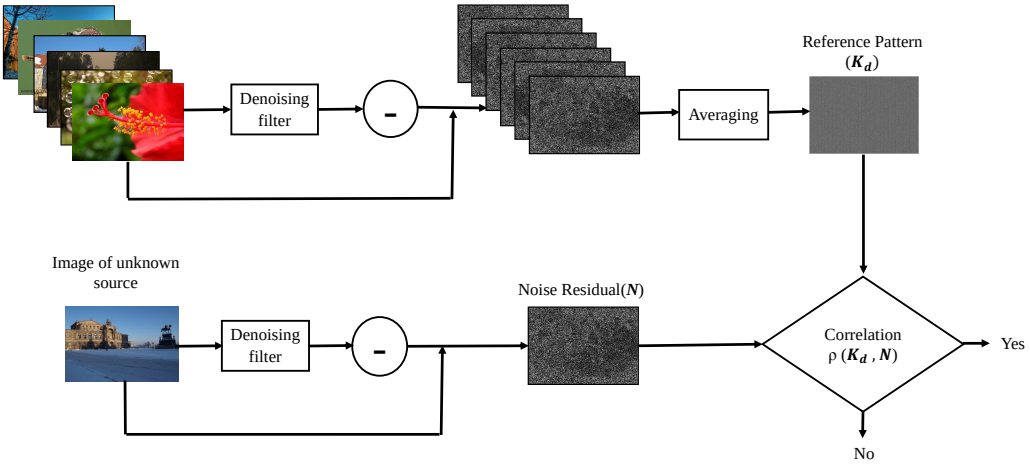


FIGURE 4.1: The correlation based scheme

All the steps used for camera identification based on the PRNU are mentioned in subsection 2.6.1.1. Figure 4.1 illustrates the steps of the correlation based method.

4.3 Proposed Machine Learning Feature Based Method

Camera identification approach based machine learning is used to classify the camera models, based on discriminant features extracted from images. In our approach we extract the features directly from what we called the POL-PRNU. The scheme presented in Figure 4.2 shows the functional diagram of our proposal. In general, the image is decomposed into its three color channels (R, G, B). The POL-PRNU of the image is obtained by applying a wavelet denoising filter and subtracted from original image. Three sets of features are extracted from POL-PRNU for classification. The following subsection describes the theoretical aspects of the POL-PRNU concept followed by the features sets details of our approach.

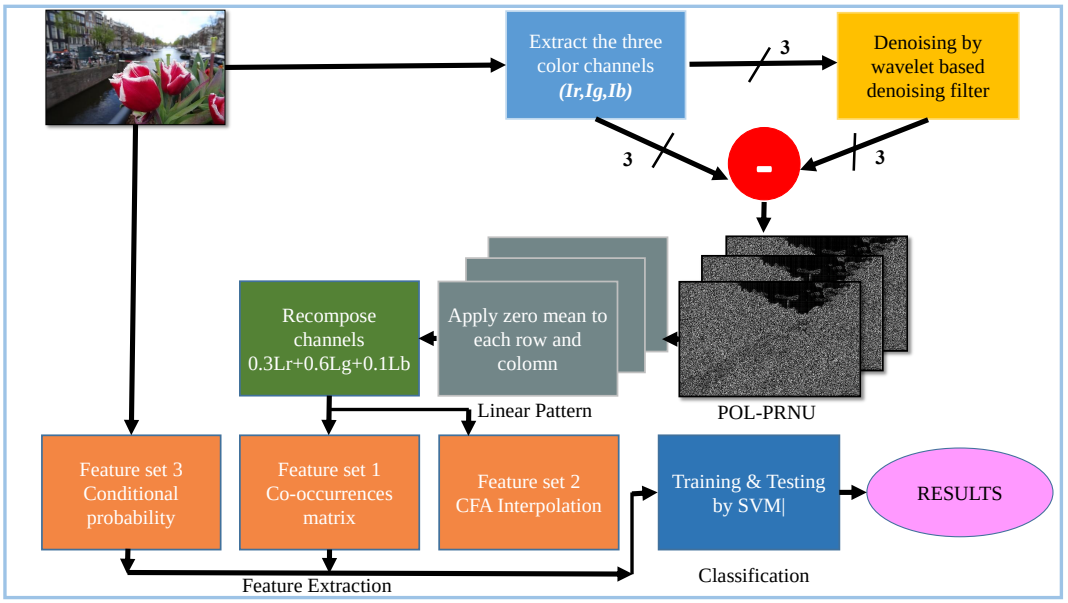


FIGURE 4.2: The proposed system framework

4.3.1 Polluted sensor noise extraction

The polluted PRNU, that we called POL-PRNU, is the sensor noise polluted by some residuals content of the image. In our approach the polluted PRNU is extracted from a single image without collecting several images to perform an averaging and extract the device reference. This leads to an easy way to extract

statistics from an image (co-occurrences and color dependencies from the polluted PRNU).

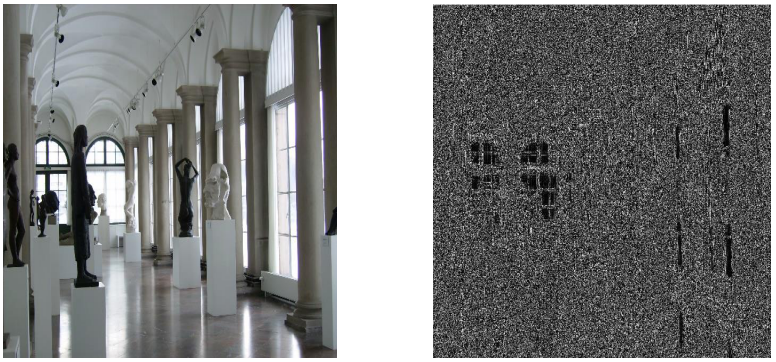


FIGURE 4.3: Example of a denoised image

Our POL-PRNU, N, is extracted by subtraction the image I and its denoised version as in Equation 2.2. For the denoising process, a wavelet based denoising filter, $F(I)$, is used based on a Wiener filtering of each wavelet sub-band for each channel as in [LFG06].

In order to suppress all artifacts introduced by color interpolation and JPEG compression, a periodic signal of pattern noise, called the linear pattern L, is extracted by subtracting the average row (respectively average column) from each row (respectively column) of N from each color channel separately [Fri09]. This leads to the three linear patterns corresponding to each color channel, noted L_r for red channel, L_g for the green channel, and L_b for the blue channel.

Finally, the three linear patterns are combined into one pattern, noted Ł, by using the conversion formula from RGB to gray-scale as in Equation 4.1. Extracting the features from the recombined fingerprint will be more reliable due to the fact that the three linear patterns are highly correlated and provide a compact information for the classifier [Fri09].

$$Ł = 0.3.L_r + 0.6.L_g + 0.1.L_b. \tag{4.1}$$

Three sets of features will be extracted, spatial co-occurrences matrix (Subsection 4.3.2), color dependencies (Subsection 4.3.3), and frequential conditional probability (Subsection 4.3.4). Co-occurrences matrix will be extracted from L_P by calculating the different statistical relationships among neighboring pixels. The second features set, related to CFA arrangement, calculates the local dependencies and periodicity among neighboring pixels. The third features set is the conditional probability features computed on the DCT domain which will be calculated from the original images by examining the absolute values of three selected coefficients in 8×8 DCT block. The following three sub-sections describe the theoretical part of the three features sets.

4.3.2 Feature set 1: Co-occurrences Matrix

The rich models for steganalysis [FK12] is adapted to extract the co-occurrences matrix of a POL-PRNU image. Rich models provide a good model for forensics applications, especially, in forgery detection and localization [MPSV15, QLLH14, VCP14]. Indeed co-occurrences is a very good way to describe the statistics of some data owning neighborhood relations, which is the case for POL-PRNU images. Calculating the co-occurrences allows dimension reduction of the POL-PRNU and gives a good representation of the statistical properties of the residual.

The co-occurrences feature vector is made of joint probability distributions of neighboring residual samples. In our case, the residual is the POL-PRNU. We use four-dimensional co-occurrence matrices formed by groups of four horizontally and vertically adjacent residual samples after they were quantized and truncated as follows:

$$R \leftarrow trunc_T(round(Ł/q)), \tag{4.2}$$

where $trunc_T$ is a function reducing the residual range with T an integer such that $T \in \{-T, ..., T\}$, $round(x)$ gives the nearest integer value of x, Ł is the linear pattern of the POL-PRNU given in Equation 4.1, $q \in \{1, 1.5, 2\}$ is the quantization step, and R is the obtained matrix.

The co-occurrences matrix will then be constructed from horizontal and vertical co-occurrences of four consecutive values $(d1, .., d4)$ from R; see equation 4.3. The horizontal co-occurrence matrix C_d^h is computed as follows:

$$C_d^h = \frac{1}{Z}\Big|\{(i,j) \mid R_{i,j} = d_1, R_{i,j+1} = d_2, R_{i,j+2} = d_3, R_{i,j+3} = d_4\}\Big|, \tag{4.3}$$

where Z is the normalization factor, with $R_{i,j} \in \mathbb{N}$ is a coefficient of the matrix R at position $(i,j) \in \{1, ..., n\}^2$, and $d = (d1, ..., d4) \in \{-T, ..., T\}^4$ with $T = 2$ in our case. Equivalently, we can compute the vertical co-occurrences matrix. This set results in **10764** features.

4.3.3 Feature set 2: Color Dependencies

The underlying assumption is that CFA interpolation algorithms leave correlations across adjacent pixels of an image. In digital cameras, the color filter array is placed before the sensor to produce a colored raw image. The CFA is usually periodic and form a certain pattern. The missing color components are then interpolated in an additional processing step using existing neighbor color components. The CFA pattern and the colors interpolation is an important characteristics of the device and can be used in the camera identification process [BSM06].

In this section, we will explain the set of features related to CFA arrangement. From the linear patterns of the noise residual L_r, L_g, and L_b, we compute local dependencies and periodicity among neighboring samples. The normalized cross-correlation is computed between the estimated linear pattern from the noise residual of the three color channels and their shifted version as in [FFG08].

For each color channel pair $(A1, A2)$, $A1, A2 \in \{L_r, L_g, L_b\}$ and shifts $\triangle_1 \in \{0, ..., 3\}$, and $\triangle_2 \in \{0, ..., 3\}$, the normalized cross correlation between two matrices is defined as:

$$\rho(A1, A2, \triangle) = \frac{\sum_{i,j}(A1_{i,j} - \overline{A1})(A2_{i-\triangle_1, j-\triangle_2} - \overline{A2})\text{ffl}}{\sqrt{\sum_{i,j}(A1_{i,j} - \overline{A1})^2 \sum_{i,j}(A2_{i-\triangle_1, j-\triangle_2} - \overline{A2})\text{ffl}^2\text{ffl}}}, \tag{4.4}$$

where ρ is the normalized cross correlation, $\triangle = [\triangle_1\ \triangle_2]^T$ is the 2D shift, $\overline{A1}$ and $\overline{A2}$ are sample means calculated from matrices $A1$ and $A2$ respectively. This step results in **96** features which are the result of six combinations of color channels by 4×4 shifts of $\triangle_1$ and $\triangle_2$.

4.3.4 Feature set 3: Conditional Probability

Conditional probability features (CP), in the frequency domain, were introduced for camera identification by Wahab et al. [AHL12]. A number of CP features can be obtained by examining the absolute values of three selected block DCT coefficients at different locations. For the usual 8×8 DCT transform, we picked three DCT coefficients from the 4×4 left upper sub-block because most non-zero coefficients are in that region. Given the three relative positions r, s, and t in a DCT block such that $\{r, s, t\} \in \{1, ..., 4\} \times \{1, ..., 4\}$, we compute the conditional probability as follows:

$$Prob(Y_i|X_j) = \frac{Prob(X_iY_j)}{Prob(X_i)}, \tag{4.5}$$

knowning that $X_i \in \{X_1, X_2, X_3\}$ and $Y_j \in \{Y_1, Y_2, Y_3\}$ are defined such as:

$$\begin{aligned} X_1 &= \{\text{value at position } r < \text{value at position } s\}, \\ X_2 &= \{\text{value at position } r > \text{value at position } s\}, \\ X_3 &= \{\text{value at position } r = \text{value at position } s\}, \end{aligned} \tag{4.6}$$

$$\begin{aligned} Y_1 &= \{\text{value at position } t < \text{value at position } s\}, \\ Y_2 &= \{\text{value at position } t > \text{value at position } s\}, \\ Y_3 &= \{\text{value at position } t = \text{value at position } s\}, \end{aligned}$$

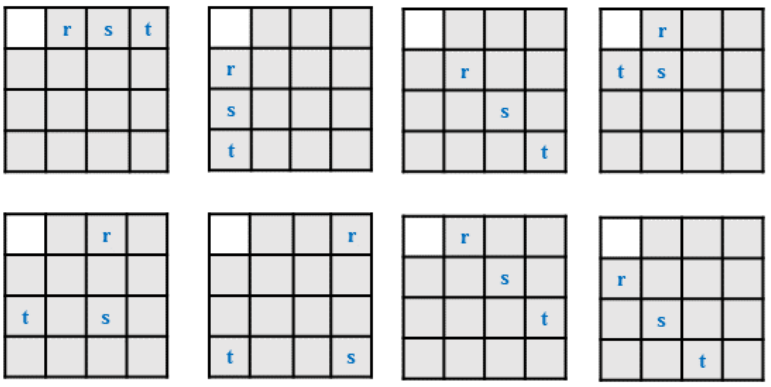


FIGURE 4.4: Eight different arrangements of r, s, t coefficients

Eight different arrangements of r, s, and t as shown in figure 4.4 will be examined over nine events resulting in **72** features.

4.4 Experimental Results

The essential step in our experiments is to extract POL-PRNU from all images. Two sets of features are extracted from POL-PRNU of each image. The third feature set is extracted from the original images. We carried out a set of camera models from Dresden database [GB10].

4.4.1 Dresden image database

Dresden image database [GB10] is one of the most widespread database dedicated to forensics applications. This is designed to fill the needs for digital image forensics by providing a useful resource for investigating camera-based image forensic methods.

Dresden database provides 16,000 authentic digital full-resolution natural images in the JPEG format, and of 1,500 uncompressed raw images. It covers different camera settings, environments and specific scenes facilitate rigorous analyses of manufacturer, model or device dependent characteristics and their relation to other influencing factors.

In our experiments, 14 different camera models are used, as shown in Table 4.1. A set of 100 images for the training and a another set of 100 images for the test are selected randomly from Dresden database for each camera model. As a result 1400 images for training and an equivalent number of images are used for testing the 14 camera model. Images for those 14 camera models are JPEG format compressed with quality factor.

Abbreviations	**Brand**	**Model**	**Resolution**
(A1)	Agfa Photo	DC-733s	3072x2304
(A2)	Agfa Photo	DC-830i	3264x2448
(A3)	Agfa Photo	Sensor 530s	4032x3024
(C1)	Canon	Ixus 55	2592x1944
(F1)	Fujifilm	FinePix J50	3264x2448
(K1)	Kodak	M1063	3664x2748
(N1)	Nikon	D200 Lens A/B	3872x2592
(O1)	Olympus	M1050SW	3648x2736
(Pa1)	Panasonic	DMC-FZ50	3648x2736
(Pr1)	Praktica	DCZ 5.9	2560x1920
(Sa1)	Samsung	L74wide	3072x2304
(Sa2)	Samsung	NV15	3648x2736
(So1)	Sony	DSC-H50	3456x2592
(So2)	Sony	DSC-W170	3648x2736

TABLE 4.1: Camera models used from Dresden database.

4.4.2 Experimental setting

The data set shown in Table 4.1 is used with the three proposed features sets. An image is decomposed into its three color channels (R, G, B). Three sets of features are extracted from noise residuals of each image, the first set is the co-occurrences vector which consists of 10764 features of different statistical relationships among neighboring pixels. The second set consists of 96 features from normalized cross correlation between POL-PRNU and its shifted versions to get the CFA interpolation dependencies among neighbor pixels. The third features set is extracted by computing the conditional probability of the 8×8 DCT transform coefficients of the original images and resulting in 72 features, see Section 4.3. This resulting in 10932 as a total number of features.

For the feature normalization step, we used the method of min-max scaling for both training and testing sets. In this approach, the features will be re-scaled to a

specific range [0,1]. This will avoid attributes in greater numeric ranges dominating those in smaller ranges. A Min-Max scaling is given by the following formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.7}$$

For the classification, LIBSVM package was used [CL11] with the Radial Basis Function (RBF) and v-fold cross validation scheme. Although SVM is a binary classification model, LIBSVM package performs multi-classification by using one-versus-rest(OVR) approach.

We used the kernel parameter $\gamma = 2^{-7}$ and cost parameter $C = 4096$ for the SVM after examining a grid search over a range of values. For $\gamma \in \{2^3, 2^2, 2^1, ..., 2^{-15}\}$ and $C \in \{2^{15}, 2^{14}, 2^{13}, ..., 2^{-5}\}$ as is recommended in [HCL03]. The training and testing sets consisted of 100 images each for each camera model. We run the training procedure 10 times then averaging the results. Each time, the training and testing data sets are selected randomly.

The proposed method achieved a total identification accuracy of 98.75% over 14 camera models as shown in Table 4.2. We recorded a perfect identification for $Agfa-Sensor530s$ and very high performance for $Canon-Ixus55$, and $Kodak-M1063$. The two models of Sony recorded the lowest rates due to the similar in-camera processes they achieve [KG15].

Camera Model	A1	A2	A3	C1	F1	K1	N1	O1	Pa1	Pr1	Sa1	Sa2	So1	So2
Correlation method%	98	98	100	96	99	98	97	100	96	98	97	96	97	95
Proposed method%	99.3	98.6	100	99.9	98.7	99.9	98.1	98	99.6	98.2	99.4	98.9	97.7	96.2

TABLE 4.2: Identification accuracy of the proposed method and the correlation based method for 14 chosen camera models.

In order to test the effect of each feature set, they are performed separately.

The first set of features of co-occurrences is chosen and performed alone. The experiment resulted in 96.91% as average accuracy. This proves the potential role of the statistical features represented by co-occurrences matrix.

Camera identification method	Accuracy
CFA features	86.93%
Co-occurrences	96.91 %
Feature based method [FFG08]	88.23%
Correlation method	97.5%
CFA+Co-occurrences	97.81%
Proposed(CFA+Co-occurrences+CP)	**98.75%**

TABLE 4.3: Results of the proposed method with comparison to another methods.

Another experiment is performed by taking the second set of features alone which it is the CFA interpolation. The last gave a result of 86.93% of accuracy. This is considered acceptable but not enough, and still less than the result of the first experiment of co-occurrences alone.

We gathered the two feature sets and implemented them together to achieve 97.81% as an average accuracy. While all the three sets achieved 98.75%. The Table 4.3 shows all the mentioned experiments with their accuracy rates.

4.4.2.1 Comparison with another feature based method

Filler et al [FFG08] proposed a method for camera model identification which aims to classify camera models by extracting some features from fingerprint. We implemented this method for comparison purposes on the same set of images from Dresden database. The later method [FFG08] proposed features are concerning statistical moments, cross correlation between color channels, block covariance, and cross correlation of linear pattern which do not describe the statistic relations of adjacent pixels.

The method in [FFG08] under comparison is tested under a similar conditions. This method only achieved 88.23% as an average identification accuracy as shown in Table 4.3. This lower result because it does not take enough descriptive features of the fingerprint.

We conclude that our method always performs better than the compared method. This is due to the strength of the descriptive features of the co-occurrences, and the additional interesting features of CFA interpolation characteristics.

4.4.2.2 Comparison with Correlation based method

For comparison, we implemented the method of the correlation based sensor pattern noise for camera identification, explained in Section 4.2. This method extracts the fingerprint of the camera which can be estimated by averaging a set of images. Normalized correlation is applied between the fingerprint and an image under test to investigate whether this image comes from this camera or not. For each camera model, we used 100 images to estimate the fingerprint and we left the rest 100 images for the test. This results in 97.5% as an average identification accuracy as in Table 4.2.

The bar chart in Figure 4.5 is showing the comparative accuracy for the two methods for each camera model separately. Only two cameras are better identified with the correlation based method (Fujifilm-FinePixJ50 with 99% compared to 98.7% and Olympus-M1050SW with 100% compared to 98%). Nevertheless, see Table 4.3, in average we can see that the proposed method performs higher than the correlation based method since it achieves 98,75% while the compared method only achieves 97.5%.

4.4.2.3 Robustness test against the overfitting

A Support Vector Machine (SVM) constructs a hyperplane, or set of hyperplanes, in a high or infinite dimensional space, which can be used for classification. The effectiveness of SVM depends on the selection of kernel function, and the kernel's parameters [HCL03].

Using a kernel function provides a single point for the separation among classes. The radial basis function (RBF), which is commonly used, maps samples into a

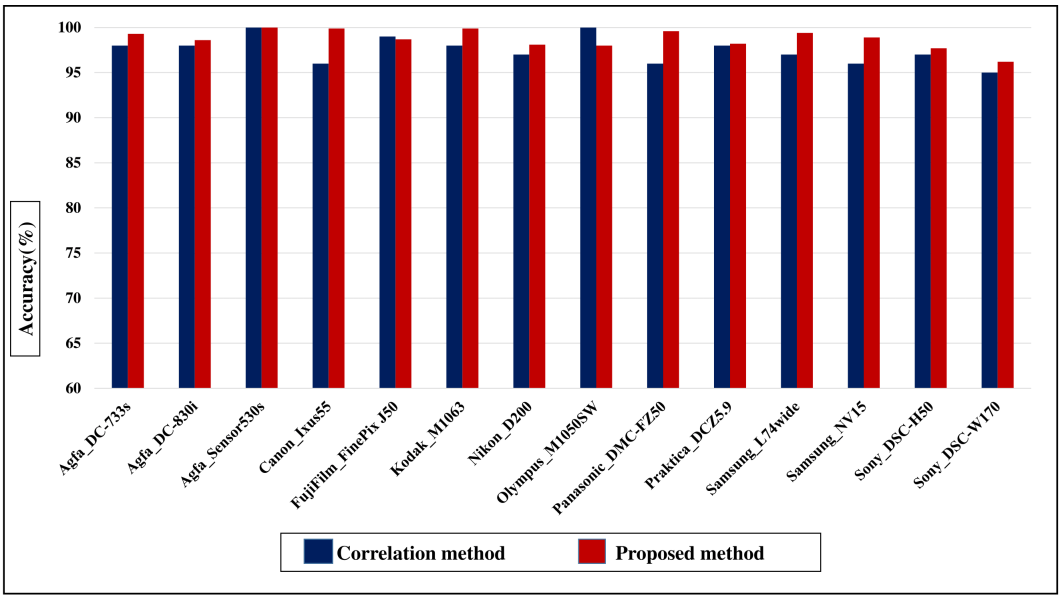


FIGURE 4.5: Comparison of the identification results.

higher dimensional space that can handle the case when the relation between class labels and attributes is nonlinear.

Projecting into high-dimensional spaces can be problematic due to the so-called curse of dimensionality. As the number of variables under consideration increases, the number of possible solutions also increases exponentially. The result is that the boundary between the classes is very specific to the examples in the training data set. The classifier has to handle the overfitting problem, so as it has to manage the curse of dimensionality [YON05].

In our case, the training and testing sets have 100 instances each, and the number of features is 10932 which is considered much larger than the number of instances. Here, we have to proceed to the learning process with a small data base and large dimension. Thus, the overfitting problem and the curse of dimensionality are occurring.

Fortunately, when the SVM uses the cross validation procedure, the cost parameter that control the over/under-fitting, is set to a value that allows a better handling of the problem and then, can prevent the overfitting problem.

In order to prove the generality of the proposed method, we performed an additional experiment. Two image subsets were downloaded from Internet database **Flickr**. The images of two camera models $Canon - Ixus - 55$, and $Fujifilm - FinePix - J50$ are used only to test the method which was trained with the previous data set **Dresden**. We achieved an identification accuracy 99.1%, 98.7% respectively as shown in Table 4.4. These results show the robustness of the proposed method since the results are similar to those obtained with Dresden (respectively 99.1% and 98.7%).

Camera Make/Model	No.Images	Identification
Canon IXUS 55	97	99.1%
Fujifilm FinePix J50	74	98.7%

TABLE 4.4: Test results for images from Flickr data set.

4.5 Conclusion

This chapter contributes in identifying camera models based on feature extraction and machine learning. The objective in adding a big number of features is to allow enhancing the identification rate by providing strong statistic tool.

The algorithm is composed of extracting three sets of features. The noise residual is obtained by applying wavelet denoising filter. Images from 14 camera models were used from the Dresden database and classified by a SVM classifier.

The experimental results show that the proposed method gives very high identification accuracy since it provides an identification rate of 98.75% in comparison with the correlation based method which achieved 97.5%. Testing images from another database proves the generality of the proposed method.

Chapter 5

Camera Model Identification Based on a CNN

Contents

5.1 Introduction

The general focus of machine learning is the representation of the input data and the generalization of the learning patterns. Good data representation can lead to high performance. Thus the key point is to construct features and data representations from raw data. Unfortunately, feature design consumes a large portion of the effort and is typically domain specific.

Deep Learning algorithms (for example CNNs) are one of the promising research fields into the automated extraction of complex data representations at high levels of abstraction. A key benefit of deep learning is that the analysis and learning of massive amounts of data make it a valuable tool for Big Data Analysis. Thus, deep learning often produces good results [NVK+15]. Nevertheless, we must say that deep learning approaches require high computing resources compared to more traditional machine learning approaches. Indeed it necessitates a powerful GPU and a big database.

Using a CNN as a black box leads to a weak performance in identifying camera model. Thus in this chapter, we evaluate the obtained gain to modify the AlexNet CNN model proposed by Krizhevsky [KSH12] which is illustrated in Figure 3.2. We also experimentally compare our CNN model to AlexNet [KSH12], and to GoogleNet [SLJ+15].

This chapter is organized as follows. Section 5.2 presents all the details of the proposed CNN architecture for camera model identification. The details of the used database come in Section 5.3. System requirements are described in Section 5.4. Section 5.5 describes the experiments and results. Section 5.6 shows a comparison with AlexNet and GoogleNet. Conclusion comes in Section 5.7.

5.2 The Proposed CNN Design for Camera Model Identification

The framework of our proposed model is shown in Figure 5.1. The first layer is the filter layer, followed by three convolutional layers from the first (Conv1) to the third (Conv3). While the last three layers are the fully-connected layers (FC1, FC2, FC3) for the classification. The details of our CNN model is illustrated in the following subsections.

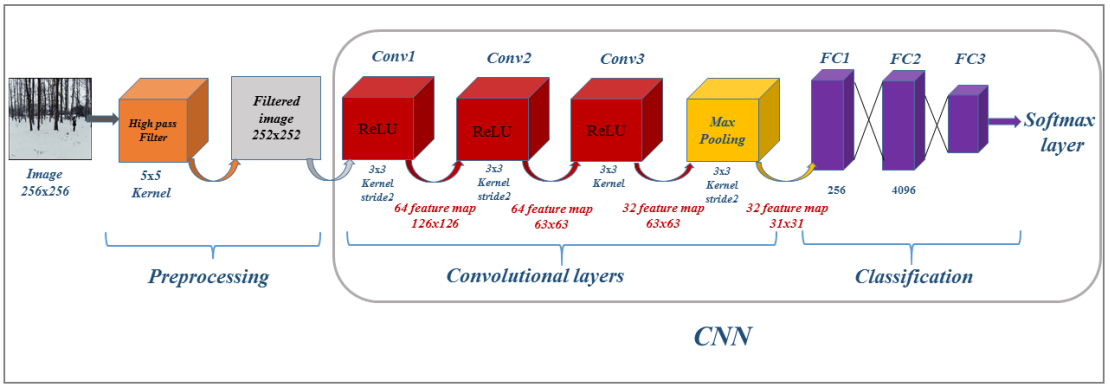


FIGURE 5.1: The layout of our Conventional Neural Networks for Camera Model Identification.

5.2.1 Filter layer

The classical way for denoising an image is to apply a denoising filter. In our experiments, we examined two types of filters as a preprocessing step, as shown in figure 5.1. The first one is the high pass filter adopted by Qian et al [QDWT15], see Equation 5.1. Applying this type of filter is important in the proposed method since it can suppress the interference caused by image edges and textures in order to obtain the image residual as follows:

$$A = I * \frac{1}{12} \begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \tag{5.1}$$

The second filter is the well known wavelet based denoising filter [Fri09]. Such a filter is applied on each color channel separately. The output of this step will fed the CNN.

5.2.2 Convolutions

AlexNet Convolutional Neural Networks [KSH12] is adapted and modified to fit the model requirements. The input to the CNN is an image of size 256×256 which will be treated by the high pass filter to produce a residual image of size 252×252. The first convolutional layer (Conv1) treats the residual image with 64 kernels of size 3×3. The size the feature maps produced is 126×126. We used the *stride* parameter equal 2 which divide the feature map size by 2. Then the second convolutional layer (Conv2) takes the output of the first layer as input. It applies convolutions with kernels of size 3×3 and produces feature maps of size 63×63. The third convolution layer applies convolutions with 32 kernels of size 3×3. The Rectified Linear Units (ReLUs) is applied to the output of each convolutional layer. ReLUs is a classical non linear function and it can lead to fast convergence with large models trained on large datasets [KSH12].

The third convolutional layer is followed by a max pooling operation with a window of size 3×3 , which operate on the feature map, and lead to a decreasing of the spatial resolution.

5.2.3 Fully Connected layers

The fully-connected layers (FC1) and (FC2) have 256, and 4096 neurons respectively. ReLUs activation function is applied to the output of fully connected layer. Each of (FC1) and (FC2) are dropped out during the learning. The output of last fully connected layer (FC3) is fed to a softmax function. The softmax function is the gradient-log-normalizer of the categorical probability distribution. It is also called the *Logistic Regression*. The softmax classifier is used in various probabilistic multiclass classification methods [Bis06]. Logistic regression is a probabilistic, linear classifier. It is parameterized by a weight matrix W and a bias vector b. Classification is done by projecting an input vector x onto a set of hyperplanes, each of which corresponds to a class. The distance from the input to a hyperplane reflects the probability that the input is a member of the corresponding class [Bis06] as detailed in Equation 5.2.

$$f_i = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}} \tag{5.2}$$

5.3 Dataset organizing

For the evaluation of the experiments, we used 33 camera models from two different data sets. The first set is made of 27 camera models from Dresden database [GB10], and the second set is made of 6 personal camera models. The list is given in Table 5.1. Using such different data sets ensure the diversity in the used data base.

Before any further manipulation, The data set is subdivided into training and testing sets, such that 80% of the data set is chosen for the training and the rest 20% for the testing data.

In order to fit the CNN model conditions, we sub-divided the chosen data set images into 256×256 and we ignored those of less than 256×256. By applying

Seq.	Brand	Model	Original Resolution	No. images 256×256
1	Agfa Photo	DC-733s	3072x2304	30349
2	Agfa Photo	DC-830i	3264x2448	39204
3	Agfa Photo	Sensor 530s	4032x3024	55585
4	Canon	Ixus 55	2592x1944	15680
5	Fujifilm	FinePix J50	3264x2448	22680
6	Kodak	M1063	3664x2748	64960
7	Nikon	D200 Lens A/B	3872x2592	55800
8	Olympus	M1050SW	3648x2736	28560
9	Panasonic	DMC-FZ50	3648x2736	37100
10	Praktica	DCZ 5.9	2560x1920	14630
11	Samsung	L74wide	3072x2304	24948
12	Samsung	NV15	3648x2736	30380
13	Sony	DSC-H50	3456x2592	36920
14	Sony	DSC-W170	3648x2736	28700
15	Agfa Photo	DC-504	4032x3024	10074
16	Agfa Photo	Sensor505-x	2592x1944	12040
*17	Canon	EOS-1200D	3648x2736	26780
*18	Canon	PowerShot SD790 IS	3648x2736	30016
19	Canon	Ixus70	3072x2304	20196
20	Canon	PowerShotA640	3648x2736	26320
*21	Canon	EOS7D	3648x2736	9360
22	Casio	EX-Z150	3264x2448	19548
23	Nikon	CoolPixS710	4352x3264	37944
24	Nikon	D70	3008x2000	13860
25	Nikon	D70s	3008x2000	13706
*26	Nikon	D5200	3648x2736	34500
27	Pentax	OptioA40	4000x3000	27885
28	Pentax	OptioW60	3648x2736	26880
29	Ricoh	GX100	3648x2736	26880
30	Rolli	RCP-7325XS	3072x2304	21384
*31	Sony	DSC-HX50	3648x2736	15960
*32	Sony	DSCHX60V	3648x2736	44400
33	Sony	T77	3648x2736	25340

TABLE 5.1: Camera models used in the experiments, models marked with * comes from personal camera models while all the others are from Dresden database.

the images sub-division step, we obtain a bigger data set which is beneficial for the training process. When doing the training/testing subdivision into two sets, we make sure that different parts of the same original image do not belong, in the same time, to the training and testing sets. Table 5.1 shows all camera models with their number of images.

5.4 System requirements

The experiments are done with a single GPU card of type GeForce GTX Titan X manufactured by Nvidia, and DIGITS training system. Many experiments were done to achieve the design of the CNN model. We measure the efficiency of the CNNs by looking at the minimum error rate after convergence. Our CNN model is shown in Figure 5.1 and detailed in Section 5.2. For each experiment, the data set is chosen randomly and the results are averaged after running the procedure 5 times with 5 different splitting of the database. By applying two different filters, explained in subsection 5.2.1, we have two different residuals which are referred to as $Residual1$ (high pass filter), and $Residual2$ (Wavelet denoising [Fri09]) in our experiments.

5.5 Experiments and Results

First, we used the first 12 camera models given in Table 5.1. For each image in the data set, a $residual1$ is extracted by applying a high pass filter [QDWT15]. Our CNN model is trained on the resulted residuals of the 12 camera models. Then we use the CNN to identify the source camera model of each image in the test set to obtain the identification accuracy. The confusion matrix is shown in Table 5.2. The average accuracy achieved by this experiment is 98%. From Table 5.2, we can see that the best identification accuracy is recorded for the camera model $Kodak-M1063$ which achieves 99.89%. $Agfa-Sensor-530s$, $Canon-55$, $Fujifilm-FinePix-J50$, $Panasonic-DMC-FZ50$, and $Samsung-L74wide$ also achieved almost perfect accuracy rates. While $Praktica-DCZ5.9$ recorded the least accuracy rate which is 90.44%.

The experiment is re-performed on the first 14 camera models of Table 5.1, by adding $SonyDSC-H50$ and $SonyDSC-W170$ to the previous 12 models. This experiment leads to 97.09% as an average identification accuracy. The total identification accuracy is shown in Table 5.4. The identification accuracy decreased with

Camera Model		1	2	3	4	5	6	7	8	9	10	11	12
Agfa DC-733s	1	96.35	1.87	-	-	-	-	-	-	-	0.61	0.92	-
Agfa DC-830i	2	2.54	94.5	0.2	0.2	-	-	0.25	-	0.21	1.69	-	-
Agfa Sensor 530s	3	-	-	99.57	-	-	-	0.23	-	-	-	-	-
Canon Ixus 55	4	-	-	-	98.54	-	-	-	-	-	0.89	-	-
Fujifilm FinePix J50	5	-	-	-	-	98.17	-	-	-	-	-	-	0.97
Kodak M1063	6	-	-	-	-	-	99.89	-	-	-	-	-	-
Nikon D200	7	-	-	0.55	-	-	0.21	97.83	0.32	-	-	-	0.61
Olympus M1050	8	-	-	-	-	-	-	0.7	96.38	0.98	-	-	0.9
Panasonic DMC-FZ50	9	-	-	-	-	-	-	-	0.78	98.46		-	0.5
Praktica DCZ 5.9	10	3.91	2.82	-	-	-	-	0.82	-	-	90.44	-	1.83
Samsung L74wide	11	1.1	-	-	-	-	-	-	-	-	0.34	98.13	-
Samsung NV15	12	-	-	-	-	0.93	-	1.21	-	0.62	-	-	96.73

TABLE 5.2: Identification accuracy (in percentage points %) of the proposed method for *Residual*1, the total accuracy is 98%. − means zero or less than 0.1.

these two models due to the fact that the captured images from camera models of the same manufacturer are sometimes harder to separate, such as $SonyDSC-H50$ and $SonyDSC-W170$. This is due, as it has been observed in [KG15], to the strong feature similarity of some camera models from the same manufacturer.

The proposed CNN model is performed again with all the 33 camera models given in Table 5.1. We achieve 91.9% as an identification accuracy for the 33 camera models for *Residual*1. As we can see, the accuracy is decreased as the number of models is increased (98% for 12 cameras, 97% for 14 cameras, and 92% for 33 cameras), and this is a known behavior in machine learning approach, especially when increasing the number of classes [Glo12].

In order to close the discussion with our CNN, it is interesting to evaluate the influence of the pooling layer. With three convolutional layers and max-pooling, the result is 98.09% whereas with only two convolutional layers and max-pooling, the result is 94.23%. The results of adding a pooling layer to the model is resumed in Table 5.3.

The experiments reference as *Residual*2 is obtained by applying a wavelet denoising filter [LFG06] on each image in the data set, then subtract the denoised image from the original one. Residuals of the training set fed the CNN model

Proposed Method	Accuracy
Two convolutional layer without Pooling	93.88%
Two convolutional layer with max Pooling	94.23%
Three convolutional layer with max Pooling	98.0%

TABLE 5.3: results for the first 12 camera models considering the pooling layer for *Residual*1.

to perform the training process. This part achieves 95.1% as total identification accuracy for the 12 camera models which is 3% lower compared to *Residual*1. We can hypothesize that the residuals obtained from such a filter suppress too much features related to some characteristic of the acquisition pipeline of a given camera model like the CFA interpolation, or lens-aberration correction traces, and that is exactly what the CNN model need to learn about the camera model features. This experiment achieved 97.09%, and 93.23% as a total identification accuracy for *residual*1, and *residual*2 respectively. The total identification accuracy is shown in Table 5.4. The results for the three data sets of camera models (12,14,33) are shown in Table 5.4.

5.6 Comparison with AlexNet and GoogleNet

AlexNet was developed by Alex Krizhevsky et al. [KSH12], and GoogleNet was designed by Szegedy et al. [SLJ+15]. These two CNNs models are trained on our data sets to be compared with our proposed CNN model. The results are illustrated in Table 5.4. GoogleNet consists of 27 layers which explain the higher score it achieves. For experiment 1, with 12 camera models, AlexNet achieves 94.5%, and 91.8% for *Residual*1, *Residual*2 respectively. GoogleNet achieves 98.99%, and 95.9% for *Residual*1, *Residual*2 respectively. We achieved with 12 camera models, 98% and 95.1% for *Residual*1, *Residual*2 respectively.

The trend is similar for the experiments with 14 camera models. AlexNet achieves 90.5% (respectively 89.45%) for *Residual*1 (respectively *Residual*2). We achieve

	(1-12) models		(1-14) models		(1-33) models
Method	residual 1	residual 2	residual 1	residual 2	residual 1
AlexNet	94.50%	91.8%	90.50%	89.45%	83.5%
GoogleNet	98.99%	95.9%	98.01%	96.41%	94.5%
Proposed Net	98.00%	95.1%	97.09%	93.23%	91.9%

TABLE 5.4: Identification accuracies for all the experiments compared to AlexNet and GoogleNet.

97.09% (respectively 93.23%) for *Residual*1 (respectively *Residual*2) and GoogleNet achieves 98.01% (respectively 96.41%) for *Residual*1 (respectively *Residual*2).

We see that our proposition improves AlexNet with 7% for the 14 camera models and the efficiency is only 1% above the bigger network of GoogleNet. As a complexity measure, the time expended for training 12 camera models using our proposed CNN model is about 5 hours and a half, while the time expended for training the same set using GoogleNet is about 16 hours. The time expended by our model for testing 12 camera is about 10 minutes against 30 minutes for GoogleNet. We conclude that our CNN model has good performance for a really smaller complexity compared to GoogleNet.

We should also add that compared to the state of the art approaches based on classical feature extraction and machine learning, the obtained results are similar with our proposition in [TCC16a], and detailed in Chapter 4. The two methods are implemented in different conditions since the classical machine learning approach [TCC16a] uses the full resolution of the data set while the proposed CNN method uses images of size 256×256. GoogleNet gives similar global accuracy (98.99%) with the same set of 14 models. This is thus a good point for CNNs approaches. By achieving the perfect design of CNNs and well tuning the network we think that we can achieve more than the classical methods listed in the state of the art.

5.7 Conclusion

In this chapter, we evaluate the efficiency of using CNNs for source camera model identification based on deep learning. We tried a small net by tuning the AlexNet

model. This small network is slightly less efficient (1% to 3%) than the biggest GoogleNet model, but the computation complexity is really low.

Scalability has also been evaluated and the increase of the number of models decreases the accuracy not too drastically. Increasing the number of layers seems to be promising and future work should explore bigger networks such as ResNet of Microsoft [KXSJ15] (which consists of more than 150 layers).

Chapter 6

Conclusions and Perspectives

Contents

6.1 Conclusions

With the increasing popularity of digital media especially in imaging devices, camera identification has become an important topic in digital forensics applications. Existing methods of camera identification can be grouped in two families, the first family is based on producing a statistical proximity based model (PRNU, radial distortion). The second family is based on machine learning and feature vector extraction. This thesis motivates in two contributions studies and improve a camera model identification through machine learning approach.

The first contribution in identifying camera models based on feature extraction and machine learning. The objective in adding this big number of features is to allow enhancing the identification rate by providing strong statistic tool. The algorithm is composed of extracting three sets of features. The noise residual is obtained by applying wavelet denoising filter. Images from 14 camera models were used from the Dresden database and classified by SVM classifier. The effectiveness of the method for source camera identification, was tested on a set of images from Dresden data-base.

The experimental results show that the proposed method gives very high identification accuracy since it provides an identification rate of 98.75% in comparison with the classical correlation based method which achieved 97.5%. The problem of over-fitting was examined by performing a robustness test with images from *Flickr*. The results are 99.1%, 98.7% for $Canon-Ixus-55$, and $Fujifilm-FinePix-J50$ respectively, which are similar to those obtained with Dresden.

The second contribution evaluates the efficiency of using CNNs for source camera model identification based on deep learning and convolutional neural networks. The contribution represents a big challenge since it is quite different from exiting conventional techniques for camera identification. We tried a small net by tuning the AlexNet model. This small network is nevertheless slightly less efficient (1% to 3%) than the biggest GoogleNet model. Scalability has also been evaluated and the increase of the number of models decreases the accuracy not too drastically.

6.2 Perspectives and Open Issues

This thesis presents the camera identification forensics. Many future perspectives can be carried out in this domain in order to increase the identification performance as follows:

- One problem related to the PRNU correlation based methods is their weak detection rate if geometrical transformations such as cropping or scaling. The direct detection will not succeed because of the desynchronization introduced by additional distortion. Our future work include the consideration of the geometrical transformations problem.
- The use of a large scale database with more camera models considering the open set scenario, so as the usage of multiple devices of the same model.
- Considering the CNN approach, increasing the number of layers seems to be promising and future work should explore bigger networks such as ResNet of Microsoft [KXSJ15] (which consists of more than 150 layers). So as the study of other denoising filters and add them inside the CNN model can increase its robustness with respect to scene traces which is what the model need to learn.
- The unknown class will be one of the perspectives, as an additional class, to handle models which are not in the training set.

Chapter 7

Résumé en Francais

Contents

7.1 Introduction

L'identification d'appareil photo vise à établir un lien entre une image et son dispositif d'acquisition en exploitant les traces laissées par les différentes étapes du processus d'acquisition de l'image. L'hypothèse de base est que les contenus numériques sont entachés d'artefacts dans aux composants internes du dispositif d'acquisition. De tels artefacts sont invisibles à l'œil humain, mais ils peuvent être utilisé pour un processus d'identification.

Dans ce chapitre, nous résumons, en français, nos deux méthodes. L'identification d'appareil photo par utilisation de bruit pollué est présenté dans la section 7.2. Dans la section 7.3, nous décrivons une identification d'appareil photo par utilisation de réseaux dé neurones convolutifs. Enfin, nous concluons à la section 7.4.

La structure générale d'un interne d'appareil photo reste semblable dans tous quelque soit l'appareil. La figure 7.1 décrit la structure de base du pipeline d'acquisition d'une image au sein d'une appareil.

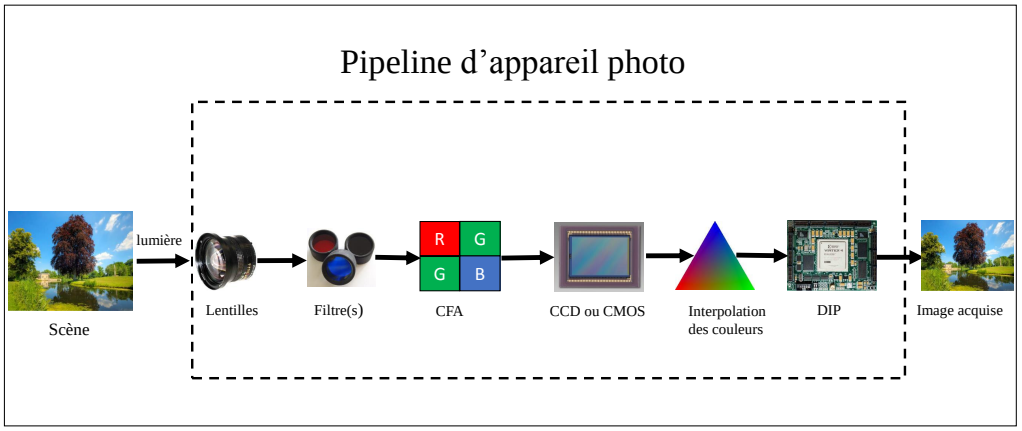


FIGURE 7.1: Pipeline d'un appareil photo

Le pipeline d'acquisition se compose d'un système de lentilles, de filtres de couleurs, d'un capteur d'imagerie et d'un processeur d'images numériques.

- Le système de lentille : Il est essentiellement composé d'une lentille et des mécanismes pour contrôler l'exposition, la mise au point et la stabilisation de l'image afin de recueillir et de contrôler la lumière de la scène.

- Les filtres optiques : Après l'entrée de la lumière dans la caméra via l'objectif, une combinaison de filtres optiques peut être utiliser comme par exemple un filtre de la lumière infrarouge).

- Le capteur d'imagerie est un réseau de rangées et de colonnes d'éléments de détection de lumière appelés photo-sites. En général, il existe deux types de capteurs de caméra déployés par des appareils photo numériques : le dispositif à couplage de charges (DCC), ou le métal-oxyde semiconducteur (MOSC). Chaque élément capteur de lumière du réseau de capteurs intègre la lumière incidente sur tout le spectre.

- Le filtre de couleur CFA : Puisque chaque élément du capteur est essentiellement monochromatique, la capture d'images en couleur nécessite des l'utilisation d'un filtre couleur CFA. Le CFA organise les pixels selon un motif, de sorte que chaque élément possède un filtre spectral différent. Par conséquent, chaque élément ne détecte qu'une bande de longueur d'onde, et l'image brute collectée à partir du capteur d'imagerie est une mosaïque de couleurs différentes et de valeurs d'intensités variables. Les modèles CFA sont généralement constitués de composantes de couleur rouge-vert-bleu (RVB).

- Opération de demosaicing : Comme chaque sous-partition de pixels ne fournit que des informations sur un certain nombre de valeurs de composantes de couleur, les valeurs de couleur manquantes pour chaque pixel doivent être obtenues par une opération de demosaicing en interpolant.

- Traitement d'image numérique : Une série de techniques de traitement d'images telles que la correction de points blancs, l'affinement d'images, la correction d'ouverture, la correction gamma, et la compression [KG15].

7.2 Identification du modèle de caméra par utiliser de caractéristiques calculées sur le bruit pollué

L'approche d'identification d'appareils photo par apprentissage automatique est utilisée pour classer les modèles d'appareils photo, en fonction des caractéristiques discriminantes extraites des images. Dans notre approche, nous extrayons les caractéristiques directement de ce que nous appelons le POL-PRNU. Le schéma présenté dans la figure 4.2 donne le diagramme fonctionnel de notre proposition.

La "PRNU pollué", que nous avons appelé POL-PRNU, est le bruit de capteur pollué par certains résidus de contenu de l'image. Dans notre approche, le PRNU pollué est extrait à partit d'une seule image.

Le POL-PRNU, noté N, est extrait par soustraction de l'image I et sa version débruité, comme appliqué dans l'équation 2.2. Pour le processus de débruitage, on utilise un filtre débruitant à base d'ondelettes, $F(I)$, basé sur un filtrage de Wiener de chaque sous-bande d'ondelette pour chaque canal [LFG06].

Pour supprimer tous les artefacts introduits par l'interpolation des couleurs et la compression JPEG, on extrait un signal périodique de bruit, appelé modèle linéaire L, en soustrayant la valeur moyenne de ligne (respectivement de colonne) de chaque ligne (respectivement colonne), pour N, pour chaque canal de couleur [Fri09]. Ceci conduit à trois modèles linéaires correspondant à chaque canal de couleur, noté L_r pour le canal rouge, L_g pour le canal vert, et L_b pour le canal bleu.

Les trois modèles linéaires sont regroupés en un seul modèle, noté Ł, en utilisant la formule de conversion de RVB vers niveau de gris comme expliqé equation 4.1. Extraire les traits de l'empreinte digitale recombinée sera plus fiable en raison du fait que les trois modèles linéaires sont fortement corrélés et fournissent une information compacte pour le classifieur [Fri09].

Trois ensembles de caractéristiques seront extraits : la matrice des co-occurrences spatiales, les dépendances des couleurs, et la probabilité conditionnelle fréquentielle. La matrice de co-occurrences sera extraite à partir de Ł en calculant les différentes relations statistiques entre pixels voisins. Le deuxième ensemble de caractéristiques, est lié à l'agencement de CFA, et calcule les dépendances locales ainsi que la périodicité entre pixels voisins. Le troisième ensemble de caractéristiques est calculé via les probabilités conditionnelles dans le domaine DCT, qui seront calculées à partir des images originales en examinant les valeurs absolues de trois coefficients sélectionnés dans le bloc 8×8 DCT.

Dans la partie expérimentale, on utilise 14 modèles d'appareils photo différents de la base de données d'images de Dresden [GB10] comme le montre le tableau 4.1. Un ensemble de 100 images pour l'apprentissage et un autre ensemble de 100 images pour le test sont sélectionnés au hasard pour chaque modèle d'appareil photo.

Le nombre total de caractéristique est 10932. Pour la classification, LIBSVM a été utilisé avec la fonction de base radiale (FBR). Le paramètre de noyau est $\gamma = 2^{-7}$ et le paramètre de coût est $C = 4096$ pour le SVM. La méthode proposée permet d'obtenir une précision d'identification totale de 98,75% comme le montre le tableau 4.2.

7.3 Identification du modèle de caméra basée sur un CNN

Dans cette section, nous proposons une méthode d'identification basé sur l'utilisation d'un CNN. Le modèle de CNN utilisé est représenté à la figure 5.1. La première couche est la couche de filtrage, suivie de trois couches de convolution (Conv1 à Conv3). Les trois dernières couches sont des couches totalement connectées (FC1, FC2, FC3) utilisés pour la classification.

Dans nos expériences, nous avons examiné deux types de filtres pour étape de prétraitement. Le premier filtre est le passe-haut adopté par Qian et al [QDWT15], comme en Equation 2.2.

La deuxième filtre est le filtre de débruitage à base d'ondelettes bien connu [Fri09]. Un tel filtre est appliqué séparément sur chaque canal de couleur.

Le modèle AlexNet [KSH12] est adapté et modifié pour s'adapter aux exigences du problème. L'entrée du CNN est une image de taille 256×256 qui sera traitée par le filtre passe-haut pour produire une image résiduelle de taille 252×252. La première couche convolutive (Conv1) traite l'image résiduelle avec 64 noyaux de taille 3×3. La taille des cartes de caractéristiques produites est 126×126. Nous avons utilisé le paramètre *stride* égal à 2 qui divise la taille de la carte de caractéristiques par 2. La seconde couche de convolutive (Conv2) prend en entrée la sortie de la première couche. Elle applique des convolutions avec des noyaux de taille 3×3 et produit des cartes de caractéristiques de taille 63×63. La troisième couche de convolution applique 32 noyaux de taille 3×3. Les fonctions d'activation utilisé sont toutes des ReLUs et sont utilisé à la sortie de chaque couche convolutionnelle. La troisième couche convolutionnelle est suivie d'une max-pooling avec une fenêtre de taille 3×3.

Les couches totalement connectées (FC1) et (FC2) ont respectivement 256 et 4096 neurones. La fonction d'activation ReLUs est appliquée à la sortie de la couche entièrement connectée. La sortie de la dernière couche totalement connectée (FC3) est passée à une fonction softmax.

7.4 Conclusion

Une image numérique peut être analysée pour identifier de l'appareil photo numérique ayant pris l'image. Dans cette thèse, deux méthodes d'identification ont été proposés.

La première méthode identifie les modèles d'appareil photo en extrayant trois ensembles de fonctionnalités dans l'approche d'apprentissage automatique. Des images de 14 modèles de caméras ont été utilisées à partir de la base de données de Dresden et classées par un classifieur SVM. Les résultats expérimentaux montrent que la méthode proposée donne une précision d'identification très élevée puisqu'elle fournit un taux d'identification de 98,75%

La deuxième contribution repose sur l'utilisation de CNN pour l'identification du modèle de source d'un appareil photo. En ajustant le modèle de AlexNet, nous obtenons un petit réseau qui est légèrement moins efficace (1% à 3%) que le plus grand modèle GoogleNet.

Chapter 8

List of Publications

8.1 List of Publications

- **[2015]**

  "**Source Camera Model Identification Using Features from Contaminated Sensor Noise**", Amel Tuama, Frederic Comby, Marc Chaumont, Chapter of Digital-Forensics and Watermarking, Springer series Lecture Notes in Computer Science, pp 83-93, Volume 9569, 31 March 2016, Revised Selected Paper from the 14th International Workshop on Digital-Forensics and Watermarking, IWDW'2015, Tokyo, Japan, October 7-10, 2015.

  Abstract: This paper presents a new approach of camera identification. It is based on using the noise residual extracted from an image by applying a wavelet-based denoising filter in a machine learning framework. We refer to this noise residual as the polluted noise (POL-PRNU), because it contains a PRNU signal contaminated with other types of noise such as the image content. Our proposition consists of extracting high order statistics from POL-PRNU by computing co-occurrences matrix. Additionally, we enrich the set of features with those related to CFA demosaicing artifacts. These two sets of features feed a classifier to perform a camera model identification. The experimental results illustrate the fact that machine learning techniques with discriminant features are efficient for camera identification purposes.

- **[2016]**

  "**Camera Model Identification Based Machine Learning Approach With High Order Statistics Features**", Amel Tuama, Frederic Comby, Marc Chaumont, EUSIPCO'2016, 24th European Signal Processing Conference 2016, Budapest, Hungary, August 29 - September 2, 2016, 978-0-9928-6265-7/16, pp 1183-1187.

  Abstract: Source camera identification methods aim at identifying the camera used to capture an image. In this paper we developed a method for digital camera model identification by extracting three sets of features in a machine learning scheme. These features are the co-occurrences matrix,

some features related to CFA interpolation arrangement, and conditional probability statistics. These features give high order statistics which supplement and enhance the identification rate. The method is implemented with 14 camera models from Dresden database with multi class SVM classifier. A comparison is performed between our method and a camera fingerprint correlation-based method which only depends on PRNU extraction. The experiments prove the strength of our proposition since it achieves higher accuracy than the correlation-based method.

- **[2016]**

  "**Camera Model Identification With The Use of Deep Convolutional Neural Networks**", Amel Tuama, Marc Chaumont, Frederic Comby, WIFS'2016, IEEE International Workshop on Information Forensics and Security, December 4-7, 2016, Abu Dhabi, UAE, 6 pages, Acceptance rate = 32%.

  Abstract: In this paper, we propose a camera model identification method based on deep convolutional neural networks (CNNs). Unlike traditional methods, CNNs can automatically and simultaneously extract features and learn to classify during the learning process. A layer of preprocessing is added to the CNN model, and consists of a high pass filter which is applied to the input image. Before feeding the CNN, we examined the CNN model with two types of residuals. The convolution and classification are then processed inside the network. The CNN outputs an identification score for each camera model. Experimental comparison with a classical two steps machine learning approach shows that the proposed method can achieve significant detection performance. The well known object recognition CNN models, AlexNet and GoogleNet, are also examined.

Bibliography

[ACC+10] Irene Amerini, Roberto Caldelli, Vito Cappellini, Francesco Picchioni, and Alessandro Piva. Estimate of prnu noise based on different noise models for source camera identification. *International Journal of Digital Crime and Forensics*, 2(2):21–33, 2010.

[AHL12] A. AbdulWahab, A.T.S. Ho, and S. Li. Inter camera model image source identification with conditional probability features. In *Proc. of the 3rd Image Electronics and Visual Computing Workshop*, 2012.

[Ale13] Piva Alessandro. An overview on image forensics. *ISRN Signal Processing*, pages Article ID 496701, 22 pages, 2013.

[AMB13] Deever Aaron, Kumar Mrityunjay, and Pillman Bruce. *Digital Image Forensics: There is More to a Picture than Meets the Eye*, chapter Digital Camera Image Formation: Processing and Storage, pages 45–77. Springer New York, New York, NY, 2013.

[AWJL11] Redi Judith A., Taktak Wiem, and Dugelay Jean-Luc. Digital image forensics: A booklet for beginners. *Multimedia Tools Appl.*, 51(1):133–162, January 2011.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Analysis and Machine Intellegence*, 35(8):1798–1828, August 2013.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[BK13] Rainer Böhme and Matthias Kirchner. Counter-forensics: Attacking image forensics. In *Digital Image Forensics*, pages 327–366. Springer New York, 2013.

[Bot12] Léon Bottou. *Stochastic Gradient Tricks*, volume 7700, page 430–445. Springer, January 2012.

[BS16] Belhassen Bayar and Matthew C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec'16, Vigo, Galicia, Spain, 2016. ACM.

[BSM06] S. Bayram, H.T. Sencar, and N. Memon. Improvements on source camera model identification based on cfa interpolation. In *Advances in Digital Forensics II, IFIP International Conference on Digital Forensics, Orlando Florida*, pages 289–299, 2006.

[Cas04] Eoghan Casey. *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, Second Edition)*. Academic Press, Elsevier, 2004.

[CFGL08] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008.

[CH04] Popescu Alin C. and Farid Hany. Statistical tools for digital forensics. In *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 128–147. Springer, 2004.

[CKLW15] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z.J. Wang. Median filtering forensics based on convolutional neural networks. *Signal Processing Letters, IEEE*, 22(11):1849–1853, Nov 2015.

[CL11] C. Chang and C. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[CLW06] K. S. Choi, E. Y. Lam, and K. Wong. Source camera identification using footprints from lens aberration. In *Proc. SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics*, volume 6069, pages 60690J–60690J–8, 2006.

[Coh07] K. Cohen. Digital still camera forensics. *Small Scale Digital Device Forensics Journal*, 1(1):1–8, 2007.

[CSA08] O. Celiktutan, B. Sankur, and I. Avcibas. Blind identification of source cell-phone model. *IEEE Transactions on Information Forensics and Security*, 3(3):553–566, 2008.

[DFR01] A road map for digital forensics research. Technical Report From the First Digital Forensic Research Workshop (DFRWS), Utica, New York, August 7-8,, 2001.

[DSM07] A. E. Dirik, H. T. Sencar, and N. Memon. Source camera identification based on sensor dust characteristics. In *IEEE Workshop on Signal Processing Applications for Public Security and Forensics, SAFE '07, Washington, USA*, pages 1–6, 2007.

[DSM08] A. E. Dirik, H. T. Sencar, and N. Memon. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Transactions on Information Forensics and Security*, 3(3):539–552, Sept 2008.

[FFG08] T. Filler, J. Fridrich, and M. Goljan. Using sensor pattern noise for camera model identification. In *Proc. ICIP, 15th IEEE International Conference on Image Processing, San Diego, California, October 12-15*, pages 1296–1299, 2008.

[FK12] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2012.

[FO01] Devernay Frédréric and Faugeras Olivier. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured enviroments. *Mach. Vision Appl.*, 13(1):14–24, August 2001.

[Fri09] J. Fridrich. Digital image forensic using sensor noise. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.

[GB10] T. Gloe and R. Böhme. The 'Dresden Image Database' for benchmarking digital image forensics. In *Proceedings of the 25th Symposium On Applied Computing (ACM SAC 2010)*, volume 2, pages 1585–1591, 2010.

[Glo12] T. Gloe. Feature-based forensic camera model identification. *Shi, Y.Q., Katzenbeisser, S. (eds.) Transactions on Data Hiding and Multimedia Security VIII. LNCS,*, 7228:42–62, 2012.

[HCL03] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[JM04] Nevine Jacob and Aline Martin. Image denoising in the wavelet domain using wiener filtering, 2004.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[Jus15] The National Institute Of Justice. Digital evidence and forensics, technical article, USA, October 28, 2015.

[KG15] M. Kirchner and T. Gloe. Forensic camera model identification. In *T. Ho, S. Li, (eds.) Handbook of Digital Forensics of Multimedia Data and Devices. Wiley-IEEE Press*, 2015.

[KMC+07] Nitin Khanna, Aravind K. Mikkilineni, George T. C. Chiu, Jan P. Allebach, and Edward J. Delp. Scanner identification using sensor pattern noise. In *Proceedings of SPIE*, 2007.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates Inc., 2012.

[KSM04] M. Kharrazi, H.T. Sencar, and N. Memon. Blind source camera identification. In *IEEE International Conference on Image Processing ICIP 04*, volume 1, pages 709–712, 2004.

[KXSJ15] He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. Deep residual learning for image recognition. *Technical Report*, 2015.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[LFG06] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on, Information Forensics and Security*, 1(2):205–214, June 2006.

[MPSV15] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva. Evaluation of residual-based local features for camera model identification. In *New Trends in Image Analysis and Processing - ICIAP Workshop : BioFor, Genoa, Italy, September 7-8*, pages 11–18, 2015.

[MS16] Owen Mayer and Matthew C. Stamm. Improved forgery detection with lateral chromatic aberration. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 2024–2028, 2016.

[MTI15] F. Marturana, S. Tacconi, and G. Italiano. A machine learning based approach to digital triage. In *In: Ho, T., Li, S. (eds.) Handbook of Digital Forensics of Multimedia Data and Devices.Wiley-IEEE Press,*, 2015.

[NVK+15] M. Najafabadi, F. Villanustre, T. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. Deep learning applications and challenges in big data analytics. *Springer*, 2, 2015.

[PPIC16] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch. *EI'2016, in Proceedings of Media Watermarking, Security, and Forensics, Part of International Symposium on Electronic Imaging, San Francisco, California, USA, 14-18 Feb*, 2016.

[PZ14] Fei Peng and Die-Lan Zhou. Discriminating natural images and computer generated graphics based on the impact of cfa interpolation on the correlation of prnu. *Digit. Investig.*, 11(2):111–119, June 2014.

[QDWT15] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. *Proc. SPIE*, 9409:94090J–94090J–10, 2015.

[QLLH14] X. Qiu, H. Li, W. Luo, and J. Huang. A universal image forensic strategy based on steganalytic model. In *Proceedings of the second ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec'14, pages 165–170, New York, NY, USA, 2014.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and

Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, June 7-12, pp. 1-9*, 2015.

[SWL07] Ashwin Swaminathan, Min Wu, and K. J. Ray Liu. Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106, mar 2007.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TCC16a] A. Tuama, F. Comby, and M. Chaumont. Camera model identification based machine learning approach with high order statistic features. *EUSIPCO'2016, 24th European Signal Processing Conference 2016, Budapest, Hungary, August 29 - September 2, 2016, pp 1183-1187*, 2016.

[TCC16b] Amel Tuama, Frederic Comby, and Marc Chaumont. *Source Camera Model Identification Using Features from Contaminated Sensor Noise*, chapter Digital-Forensics and Watermarking: 14th International Workshop, IWDW 2015, Tokyo, Japan, October 7-10, 2015, Revised Selected Papers, pages 83–93. Springer Publishing, 2016.

[TN08] Sencar Husrev T. and Memon Nasir. *Overview of State-of-the-Art in Digital Image Forensics.* Algorithms, Architectures and Information Systems Security, World Scientific Publishing Co., Inc., pages 325-348, 2008.

[TRC15] Thanh Hai Thai, Florent Retraint, and Rémi Cogranne. Camera model identification based on dct coefficient statistics. *Digit. Signal Process.*, 40(C):88–100, May 2015.

[TRC16] Thanh Hai Thai, Florent Retraint, and Rémi Cogranne. Camera model identification based on the generalized noise model in natural images. *Digit. Signal Process.*, 48(C):285–297, January 2016.

[Vap95] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[VCP14] L. Verdoliva, D. Cozzolino, and G. Poggi. A feature-based approach for image tampering detection and localization. In *IEEE International Workshop on Information Forensics and Security (WIFS), Georgia, USA,*, pages 149–154, 2014.

[VEK07] L. T. Van, S. Emmanuel, and M. S. Kankanhalli. Identifying source cell phone using chromatic aberration. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 883–886, July 2007.

[XS12] G. Xu and Y. Q. Shi. Camera model identification using local binary patterns. In *Proc. IEEE Int Conference on Multimedia and Expo (ICME) , Melborne, Australia*, pages 392–397, 2012.

[YON05] Bengio Yoshua, Delalleau Olivier, and Le Roux Nicolas. The curse of dimensionality for local kernel machines. Technical Report 1258, Département d'informatique et recherche opérationnelle, Université de Montréal, 2005.

[ZF14] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833, 2014.

[Zur92] J. Zurada. *Introduction to Artificial Neural Systems.* West Publishing Co., 1992.

Forensic Source Camera Identification by Using Features in Machine Learning Approach

Abstract:

Source camera identification has recently received a wide attention due to its important role in security and legal issue. The problem of establishing the origin of digital media obtained through an imaging device is important whenever digital content is presented and is used as evidence in the court. Source camera identification is the process of determining which camera device or model has been used to capture an image.

Our first contribution for digital camera model identification is based on the extraction of three sets of features in a machine learning scheme. These features are the co-occurrences matrix, some features related to CFA interpolation arrangement, and conditional probability statistics computed in the JPEG domain. These features give high order statistics which supplement and enhance the identification rate. The experiments prove the strength of our proposition since it achieves higher accuracy than the correlation-based method.

The second contribution is based on using the deep convolutional neural networks (CNNs). Unlike traditional methods, CNNs can automatically and simultaneously extract features and learn to classify during the learning process. A layer of preprocessing is added to the CNN model, and consists of a high pass filter which is applied to the input image. The obtained CNN gives very good performance for a very small learning complexity. Experimental comparison with a classical two steps machine learning approach shows that the proposed method can achieve significant detection performance. The well known object recognition CNN models, AlexNet and GoogleNet, are also examined.

Keywords: Camera Identification, PRNU, Co-occurrences, CFA interpolation, Deep Learning, Convolutional Neural Networks.

Identification d'appareils photos par apprentissage

Résumé:

L'identification d'appareils photos a récemment fait l'objet d'un grand intérêt en raison de son apport au niveau de la sécurité et dans le cadre juridique. Établir l'origine d'un média numérique obtenu par un appareil d'imagerie est important à chaque fois que le contenu numérique est présenté et utilisé comme preuve devant un tribunal. L'identification d'appareils photos consiste à déterminer la marque, le modèle, ou l'équipement qui a été utilisé pour prendre une image.

Notre première contribution pour l'identification du modèle d'appareil photo numérique est basée sur l'extraction de trois ensembles de caractéristiques puis l'utilisation d'un apprentissage automatique. Ces caractéristiques sont la matrice de co-occurrences, des corrélations inter-canaux mesurant la trace laissée par l'interpolation CFA, et les probabilités conditionnelles calculées dans le domaine JPEG. Ces caractéristiques donnent des statistiques d'ordre élevées qui complètent et améliorent le taux d'identification. La précision obtenue est supérieure à celle des méthodes de référence dans le domaine basées sur la corrélation.

Notre deuxième contribution est basée sur l'utilisation des CNNs. Contrairement aux méthodes traditionnelles, les CNNs apprennent simultanément les caractéristiques et la classification. Nous proposons d'ajouter une couche de pré-traitement (filtre passe-haut appliqué à l'image d'entrée) au CNN. Le CNN obtenu donne de bonne performance pour une faible complexité d'apprentissage. La méthode proposée donne des résultats équivalents à ceux obtenus par une approche en deux étapes (extraction de caractéristiques + SVM). Par ailleurs, nous avons examinés les CNNs : AlexNet et GoogleNet. GoogleNet donne les meilleurs taux d'identification pour une complexité d'apprentissage plus grande.

Mots clés: Identification de l'appareil source, PRNU, co-occurrences, Interpolation CFA, L'apprentissage en profondeur, Réseaux de neurones convolutif.