Digital Forensics Ballistics

Reconstructing the source of an evidence exploiting multimedia data

Author:

OLIVER GIUDICE

XXIX Ciclo di Dottorato in Matematica e Informatica
Department of Mathematics and Computer Science
UNIVERSITY OF CATANIA

Supervisor: Prof. Sebastiano Battiato

JULY 2017

Abstract

The Forensic Science (sometimes shortened to Forensics) is the application of technical and scientific methods to the justice, investigation and evidence discovery domain. Specifically finding evidences can be trivial and in many fields is achieved with methods that exploits manual processes and the experience of the forensics examiner. Though human factor can be often discussed and the evidences collected and found without repeatable and scientific methods could be of no use in tribunal. For these reasons this thesis focus on the investigation and development of classification engine able to uniquely identify and classify evidences in a scientific and repeatable way: each decision is driven by features and is associated by a confidence number that is the evidence itself. Two application in two different domains of Ballistics will be described: Image Ballistics, that is the reconstruction of the history of an image, and Fire Weapon Ballistics, that is the identification of the Weapon that fired an investigated bullet from the imprintings left on the bullet cartridge. To understand how to solve these two real in-the-field problems, multimedia-based novel techniques will be presented with promising results both in Image and Classic Ballistics domain.

Dedication and acknowledgements

During these long 3 years of PhD course, I have walked so many research paths that I do not remember all of them! I started from saliency detection techniques in video to identify human eye focus, passing through Computer Vision applications in traffic domain, passing then to techniques of Lossless compression of images finally landing to what was my love at first sight: the world of digital forensics.

Having changed frequently the research topic has definitely impacted on my final research results but has allowed me to enormously increase my cultural and scientific knowledge not only in the digital forensics field, but also in the image processing and Computer Vision applications.

A first thanks goes to my supervisor: Prof. Sebastiano Battiato, who has followed me in these three years and has allowed me to complete the PhD course although compromised by numerous interruptions and obstacles.

Interruptions and obstacles that have mainly come from the recruitment in Banca d'Italia where I currently work as an IT research staff member. Actually, I must thank Banca d'Italia for allowing me to continue studying in my PhD course.

For Banca d'Italia my special thanks go to the Applied Research Team (Giuseppe Galano, Andre Gentili, Cecilia Vincenti et al.) of the IT department of Banca d'Italia, and specifically the coordinator dr. Marco Benedetti, who has been my source of inspiration and professional growth for these years.

I'd like to thank the "Reparto Investigazioni Scientifiche di Messina" (RIS) for providing data and domain-specific knowledge without which I would not be able to get the results presented in this thesis. Likewise, I would like to thank iCTLab for providing the data for the social image ballistics study and dr. Antonino Paratore for the help during the research activity.

A final thanks to the Image Processing LAB (IPLAB) reality from which I was born professionally and whose members have accompanied me in these years.

Last but not least a thank you goes to my former girlfriend (three years ago) now my wife Cristina, to my parents and to everyone in someway has or had something to do with my life :D

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Many different fields of science are finding new application in the Forensic Science giving investigators more and more powerful tools. One of the Forensics fields that is having a lot of successful results is the Digital Forensics: Computer Science meets Forensics The earliest notion of digital forensics came when the Federal Rulers (US) first started to discuss about digital evidence in the 1970s. Real digital forensics investigations started 80s when federal agents started to take care of computers in search of digital evidences. This almost amateur approach continued until late 90s when academia researchers started to figure out that Digital Forensics is a field with problems and possibilities big enough to warrant investigation. In the final report of The first Digital Forensics Research Workshop (DFRWS), in 2001, there was a first definition of the Digital Forensic Science:

"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".

Starting from this first definition, following the evolution of the digital world, Digital Forensics has developed into new specific fields:

- *Multimedia Forensics*: Analysis of digital images and videos from digital cameras, smartphones, etc. or the web to solve a question;

- *HW/SW Forensics*: Analysis of hardware components and software applications for the detection of technical characteristics, operational potentials and operations carried out through them;

- *Network Forensics*: Analysis of internet data traffic flows and navigation meta-data and p2p platforms / systems for reconstruction of events;
- *Audio Forensics*: Voice speaker recognition, judicial transcripts. Audio enhancement;
- *Mobile Forensics*: Mobile devices (smartphones, tablets, etc.) analysis;
- *Disk Forensics*: Analysis of mass storage media and reconstruction of events (timeline) for trial purposes;

The above-mentioned list is not exhaustive and much more new fields are emerging due to technology and scientific advance. Moreover some fields are dividing and specializing. For example in the Multimedia Forensics field now covers audio, video and image forensics each sub-field with its peculiarities and specializations. The introduction and wide-spread of digital sensors have made the scenario even more complex. Sensors can capture parts of the reality and transform it into digital representations, such as images or audio files, which are then stored and processed by computers. Such digital representations can be subject to forensic investigations, but they can only serve as probative facts if they are reliable and authentic.

In this thesis the Forensics Ballistics examination task will be covered in the most general way. Ballistics can be defined indeed as the reconstruction of the history of an evidence starting from the traces of the device that fired (acquired, produced, altered) the evidence itself. Starting from this definition Ballistics can be referred both to device source identification of images and Classic Ballistics examination of firearms traces on bullets and cartridges. This two parts of the meaning of Ballistics will be analysed. At first the Multimedia Forensics techniques will be surveyed starting with special dedication to JPEG traces and an novel application to Social Networks. Then classic ballistics examination of firearms proofs will be addressed with new multimedia-derived automatic techniques.

CHAPTER 2

EXPLORING THE MULTIMEDIA FORENSICS DOMAIN

Nowadays over 4.2m CCTV cameras are just in the UK, that achieved the first position in global league table for ratio of cameras to people. All those CCTVs record an huge amount of image data representing people in everyday life. Moreover the wide spread of Social Networks, with more than 1 billion images shared each day just on Facebook, produces what today we know as "Big Data". For these reasons, a new sector of Digital Forensics has born with the name of Multimedia Forensics. The term Multimedia Forensics appeared for the first time in early 2000 [149]. Over the past couple of years, the relatively young field of multimedia forensics has grown dynamically and now brings together researcher from different communities, such as multimedia security, computer forensics, imaging, and signal processing. Techniques from multimedia forensics provide ways to test the source of digital sensor data in order to authenticate an image and to test the integrity of the image itself in order to identify forgeries. Multimedia forensics covers images, video and audio contents. More specifically, Image Forensics analyses an image, by using image processing science and domain expertise, in order to reconstruct the history of an image since its acquisition (Image Ballistics), and to detect forgeries or manipulations (Image Integrity/Authenticity). A brief survey of the most efficient and recent approaches to solve the Image Ballistics task (Section 5.1) and the Image Authentication one (Section 2.2.1) will be presented in the remainder of this chapter.

2.1 Image Ballistics

Recent developments in Image Forensics have shown that the task of identifying the type of device, the model or even the sensor used to shot a photo or a video is possible. This can be done exploiting the traces left by the acquisition sensors as fingerprints. This discovery has attracted the interest of Law Enforcement Agencies (LEAs) for its potential usefulness for investigation

purposes. The possibility to establish a connection between a photo to a device can be used to trace the author of a digital document involved in criminal activities (like in the case of images and videos portraying child pornography or terroristic activity). The above concept, usually referred to as device fingerprinting, can be extended to a big variety of other traces embedded within digital contents, for instance, when an image is uploaded to a Social Network platform or when it is processed by means of a specific processing suite. Forensic analysis of these fingerprints may empower LEAs with a new class of investigative instruments, which can be extremely useful to combat cybercrimes and gather evidence in conventional investigations. The most effective techniques for image ballistics are intended to classify images acquired with scanner devices from those generated by a digital camera or those images altered with software processing. These techniques take into consideration the acquisition pipeline in a digital camera by considering traces left by each of the acquisition steps like the aberration introduced by the lens, the type of colour interpolation, the imperfections of the sensor that captures the scenes and the jpeg format used, in terms of image compression parameters. Figure 2.1 shows the acquisition pipeline for a digital image from its acquisition phase to the last storage step. In general, given an image, image ballistics aims to recognize the class and/or individual features of the device that generated the digital image showing the origin of the image. This can be done by exploiting different techniques that will be described in the next subsections.

2.1.1 Lens Aberration Techniques

The manufacturing process of lenses used in digital cameras produce different kinds of aberrations on images. Generally two kind of lens aberration are investigated to solve the problem of source camera identification: lens radial distortion [157] and chromatic aberration [101, 168]. Most of digital cameras are equipped with lenses having almost spherical surfaces that introduce radial distortions, this is done by manufacturers to reduce production costs, The radial distortion of spherical lenses causes straight lines in the object space rendered as curved lines on camera sensor and this occurs when there is a change in transverse magnification with increasing distance from the optical axis. The degree and the order of compensation of such a distortion vary from one manufacturer to another or even in different camera models. The goal in the method proposed in [157] is to find the distortion parameters that constitute the fingerprint to identify source camera following the Devernay's straight line method [60]. 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 solve the radial distortion on an image. The second type of lens aberration is the chromatic aberration, that is the phenomenon where light of different wavelengths fail to converge to the same position of the focal plane. Specifically there are two kind of chromatic aberration: longitudinal aberration, that causes different wavelengths to focus at different distances from lens, and lateral aberration that is related to different positions on the sensor. In both cases, chromatic aberration leads to various

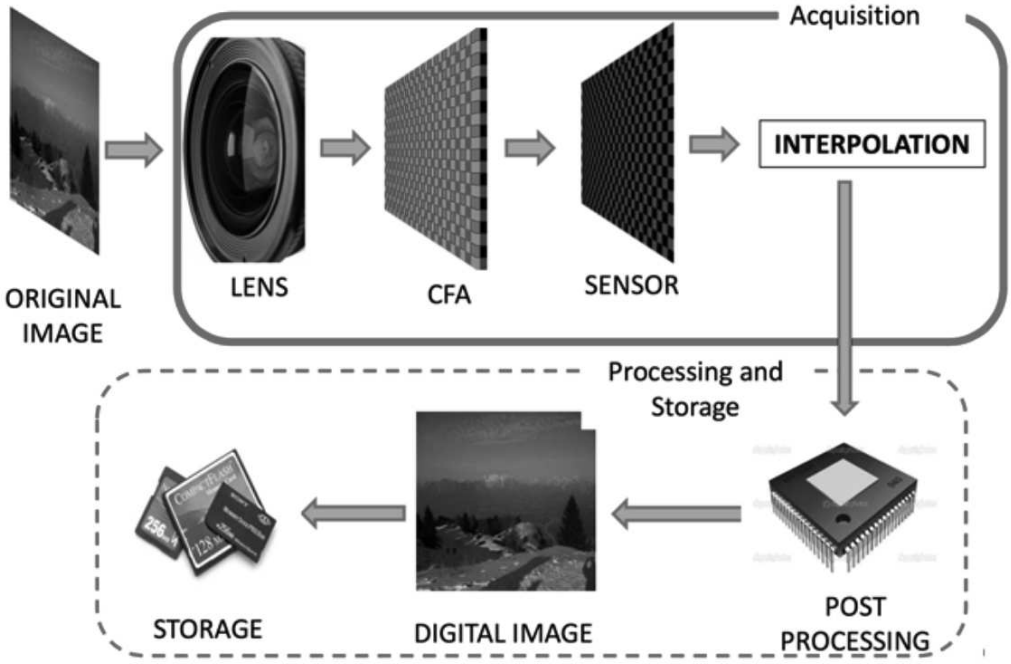


Figure 2.1: Acquisition pipeline of a digital image.

forms of colour imperfections on the final image. Only lateral chromatic aberration is taken into consideration in [168] for source identification of cell phone devices by exploiting a SVM classifier.

2.1.2 CFA Interpolation Techniques

In the acquisition pipeline of digital images, the acquisition of colours is done by decomposing the visible light into three basic components corresponding to the wavelength of red, green and blue. In theory, a different sensor for each of the colours to be captured is needed but this would raise the of price camera devices exceedingly and introduces various technical complications. Commercial devices, instead, exploit another solution: in the acquisition pipeline, before light arrives to the sensor, a thin photosensitive layer is applied, known as CFA (Color Filter Array).

The CFA filters the light by separating the three basic colours. In this way a grid of values is obtained, through which each pixel of the sensor, records the signal relating to a single colour component. The most popular type of CFA is the Bayer Pattern, represented by the multi-colour chessboard displayed in Figure 2.1. The use of the CFA reduces the number of sensors needed to one for coloured image acquisition. To do this reduction, a reconstruction of the two missing colour components by means of an algorithm of interpolation (demosaicing) is needed. Usually the processor of the acquisition device performs this process before the image is stored. As mentioned previously, it is necessary to reconstruct the signal captured by the CFA of the sensor in order to obtain an RGB image at full resolution. The brightness of missing values are obtained by

applying a procedure of the image interpolation relative to each component of R, G and B that has been extracted by the CFA. There are several interpolation methods all of them estimate the values of the missing pixels by combining those of the neighbouring pixels using an interpolating function. Given an image I, the techniques for camera source identification are focused on finding the pattern of the CFA and the colour interpolation algorithm employed in internal processing blocks of a digital camera that acquired I [29]. One well-known approach, proposed by Bayram et al. [163], estimates the colour interpolation coefficients in different types of texture of the image (smooth, horizontal gradient and vertical gradient image regions) for every CFA pattern, through a linear approximation.

2.1.3 Sensor Imperfections Techniques

This class of approaches for camera source identification exploits systematic errors introduced by sensors that appear on all images acquired by that sensor. A way to observe these imperfections is by taking a picture of an absolutely evenly lit scene, the resulting digital image will exhibit small changes in intensity among individual pixels. These errors include sensor's pixel defects and a pattern of noise. The noise pattern can be divided into two major components, namely, fixed pattern noise and photo response non-uniformity noise (PRNU). The noise is typically more prevalent in low-cost cameras and can be numerically evaluated through the analysis of multiple images obtained from the same camera. One example has been proposed by Lukas et al. [117]: for each camera under investigation, they first determine its reference pattern noise, which serves as a unique identification fingerprint. Thus to identify the camera from a given image, they considered the reference pattern noise as a spread-spectrum watermark, whose presence in the image is discovered by using a correlation detector. PRNU can be difficult to be evaluated due to image content and the number of images available from a specific camera device hence Lawgaly et al. [111] proposed a sharpening method to amplify the PRNU components for better estimation, thus enhancing the performance of camera source identification. Another recent technique of PRNU-enhancing has been proposed by Debiasi et al. [59] by parametrizing the enhancement of PRNU based on image contents.

2.1.4 Techniques based on JPEG analysis

Since the JPEG image format has emerged as a virtual standard, most devices and softwares encode images in this format. This lossy compression scheme allows for some flexibility in how much compression is achieved. Manufacturers typically configure their devices differently to balance compression and quality to their own needs and tastes [27]. This difference, embodied in the JPEG quantization tables, can be used to identify the source of image. The first work on image ballistics based on JPEG quantization tables has been proposed by Farid et al. [71] and the effectiveness of the idea was demonstrated in subsequent works like [105, 107]. Given the possibility to identify the acquisition device from JPEG quantization tables, new approaches

emerged trying to estimate the original DQTs for multiple compressed JPEG images [82, 170]. Nowadays Social Networks allow their users to upload and share large amounts of images: just on Facebook about 1 billion images are shared every day. A Social Network is yet but another piece of software that alters images for bandwidth, storage and layout reasons. These alterations have been proved to make state-of-the-art approaches for camera identification less precise and reliable. Recent studies [46, 128] have shown that, although the platform heavily modifies an image, this processing leaves a sort of fingerprint on the JPEG image format. This evidence can be exploited to understand if the image has been actually uploaded to a particular social platform. Another big family of techniques based on JPEG standard are those that exploit the meta-data stored on JPEG files. The majority of modern cameras encapsulate meta-data in JPEG files, consisting in a format called EXIF [58]. An acquisition device can save within EXIF data information such as the producer and the model of the camera, information related to the date and time of image acquisition, information on characteristics of the image (pixel resolution, dpi, colour depth, etc.), shooting settings (shutter speed, aperture, flash, focus, etc.), GPS coordinates and others. While recording so many information, image ballistics based on EXIF very weak because it cannot guarantee that all information discovered are genuine. In fact it is very easy to alterate EXIF data. In order to discover image tampering, Kee et al. [105] proposed an image authentication approach based on EXIF data.

2.1.5 Discussion

The adoption of Forensic fingerprinting techniques in real investigative scenarios is problematic for a number of issues. First, the lack of robustness in realistic operating conditions. Existing techniques exhibit high accuracy when tested under controlled conditions. However, their use in a realistic setting is problematic, since the accuracy in such conditions is either unknown or known to drop dramatically [86]. This is particularly true for videos, since the adoption of video compression schemes usually inhibits the possibility of extracting the fingerprints [32, 33]. This problem is even more serious when the evidence gathered by forensic tools must be used as proof in court. In this case, in fact, the tools should be validated through properly defined procedures, and their accuracy established according to precise models, which, up to date, have not even been defined. Another big issue is the necessity of processing huge amount of data that is crucial in several investigative scenarios. In the case of child abuse or terrorism, for instance, seized hard drives may contain thousands of images and videos that must be processed in a timely fashion. A few approaches have been proposed so far to allow the extraction and analysis of device fingerprints contained in large amounts of images [90], however such techniques have not been extensively validated on real world data and there are still several open issues regarding their scalability to huge amounts of images and their robustness to simple processing operations [167]. Moreover, the above techniques have only been tested on images, so that their applicability to digital video has yet to be investigated. Real world investigations

are characterized by the necessity of dealing with heterogeneous data stemming from diverse sources. As an example, textual metadata is often associated to multimedia contents, in the form of file headers, annotations or accompanying text, and should be used to complement the information inferable from the digital media itself. This is not an easy task since it entails a complex data fusion, multi-clue, process whose study has been undertaken only recently [53, 78]. As an additional difficulty, available information is often incomplete and unreliable. For instance, as opposed to laboratory conditions, the device classes that may have been used to produce an image are not fully known [16, 51, 97]. It is, then, necessary that proper tools to aggregate evidence obtained through different means and with different reliability, levels are developed. Finally the lack of reliability and countermeasures against deception attempts can be risky: it has been recently shown [39] that the adoption of simple counter-forensics strategies allows disabling most of the multimedia forensic techniques developed so far. Fake evidence can even be built if criminals have sufficient information about the forensic tools and the traces such tools look for. This raises several problems, especially when forensic evidence must be used in court. Anti-counter-forensics techniques developed so far are usually rather limited and address only one attack at a time, thus entering a typical cat-and-mouse loop in which attacks and countermeasures are developed in a greedy fashion. This naive approach slows down the development of general attacker-aware forensic techniques and impedes to evaluate the ultimate security of multimedia forensics tools [21].

2.2 Integrity and Authentication of Images

Back in 1946, making adjustments required a lot more than a computer, some software, and some pointing-and-clicking skills. Retouching required a whole box of tools, a very sharp eye, and an extremely steady hand. Nowadays, retouching and manipulating photographs is done with fancy photo-editing programs like Adobe Photoshop or GIMP or even in a more immediate and easy way through camera applications in mobile phones that offer capabilities of heavy image filtering, enhancing and editing. There are many modifications that can be applied to an image. They can be grouped into four categories: splicing, inpainting, enhancement and geometric transformations.

Given the huge amount of editing possibilities that can be done on an image, the question is: when retouching or image editing becomes a forgery? "Forgery" could be seen as a subjective word, an image can become a forgery based upon the context in which it is used. An image can be altered for fun or just for enhancing a bad photo. It is possible to agree that all editing that are applied to images in order to improve its appearance cannot be considered a forgery even though it has been altered from its original capture.

The detection and prevention of malicious forgeries on images, have focused the attention of Image Forensics researchers. It is possible to identify two different families of image authenti-

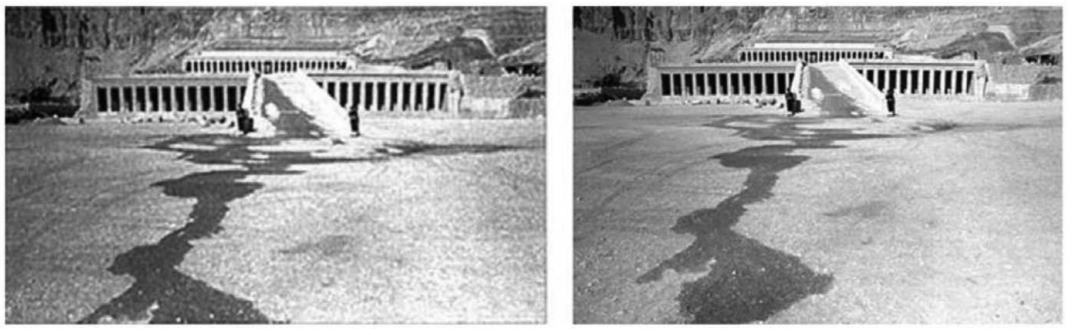


Figure 2.2: An example of image forgery that changes the content in order to make the recipient to believe that the objects in an image are something else from what they really are. In November 1997, after 58 tourists were killed in a terrorist attack at the temple of Hatshepsut in Luxor (Egypt), the swiss tabloid "Blick" digitally altered a puddle of water to appear as blood flowing from the temple.

cation approaches: active approaches, that insert or attach additional data on images in order to achieve the authentication of the image itself; passive approaches that exploit the intrinsic information of images to detect forgeries.

2.2.1 Active Image Authentication Techniques

Active image authentication approaches are based on traditional cryptography, watermarking and digital signatures based on image contents. Image authentication methods based on cryptography compute a message authentication code from images using an hash function. The resulting hash is further encrypted with a secret private key of the sender and then appended to the image. Techniques that are based on the hash computing of image lines and columns are known as line-column hash functions: separate hashes are obtained for each line and each column of an image. These hashes are stored and compared afterwards with those obtained for each line and each column of the image to be tested. If any change in the hashes is found, the image is declared manipulated and the tampered region is detected, otherwise the image is authentic [93]. Authentication methods based on watermarking consist in calculating a piece of data and hiding it in the image. The authentication is done by extracting it when it is necessary. The basic idea behind watermarking techniques is to generate a watermark and to insert it in the image in such a way that any modification made to the image is also reflected on the watermark. Simply verifying the presence of the inserted watermark allows the image authenticity verification and eventually localization of tampered regions. If the watermarking does not tolerate any image distortion it is fragile watermarking. In this case, the image is considered authentic if and only if all its pixels remain unchanged. The first algorithms of fragile watermarking has been computed from a set of image pixels. This set of pixels may be chosen with the help of a secret key. One of the first techniques that used image authentication though fragile watermarking was proposed

by Walton [171]; it used only image information to generate the watermark. This technique is based on the insertion, in the least significant bits (LSB), the checksum calculated with the grey level of the seven most significant bits of pseudo-randomly selected pixels. This method was able to detect and localize manipulations but with no restoration capabilities. This method has the advantage of being very simple and fast. Moreover, it detects and localizes tampering. However, the algorithm cannot detect the manipulation if blocks from the same position of two different images, which are protected with the same key were exchanged. To avoid this type of attack, several improvements were made to this method by extracting more robust bits [52]. Other active techniques are based on the extraction and exploitation of image high level features like SIFT that can be encoded in a codebook and attached to an image. In this way the recipient can infer if any editing is done on receiving image just by extracting the features and comparing with the original attached codebook. Battiato et al. in [23] a SIFT-based robust image alignment tool for tampering detection.

2.2.2 Passive Image Authentication Techniques

The passive authentication methods can identify if an image suffered any kind of editing without the help of the additional information. In the process of a skilled forgery, besides changing the important region about the image content, there are lots of post-processing manners that can be used to remove the artificial trace. The post-processing will make the forensics approaches less effective, so how to deal with various post-processing and improve the robustness of forensics methods has become a very important subject. To achieve the best results, state-of-the-art approaches focus on detecting specific tampering according to specific image editing categories as listed in section 2.2. The majority of studies focus on the splicing detection, known also as copy-move forgery detection. [40] is a pixel-based approach that tries to identify the tampered region by analyzing specific and deterministic transformation done on pixel-blocks taken from the image itself (e.g., translations, rotations, scale transformations). Better results are obtained with approaches based on features extracted from different color spaces [187] and exploited with an SVM classifier. Other splicing detection methods are based on incoherences discoverable in frequency domain: [129] propose a method that exploits a shift-invariant version of the discrete wavelet transform for data analysis. Finally, [135] presents good results in detecting splicing tamperings by analyzing the intrinsic noise of an image. The task of detecting a forgery on image becomes increasingly difficult, as said before, if multiple post-processing are done on an image. In this case complex editing are applied to images and the only way to detect forgeries is by analyzing geometry and physical features in the contents of the image itself. Different approaches achieve good results by analyzing the contents of images based on scene illumination [153]. image edges distribution [44] and by comparing high level features (e.g., SIFT, SURF) of genuine images with tampered ones exploiting SVM classifiers [15, 38, 134, 159]. For most tampered images, in order to hide the tampering itself, median or blurring filters are applied to the image itself.

Understanding if an image has been blurred for malicious intent can be very interesting and for this purpose. [45] and [138] are two remarkable works for automatic detection of such forgeries. Last but not least, the JPEG format stores many information not only useful for Image Ballistics but also for forgery detection. Some approaches stand above the others: a group of them looks at the structure of the file: JPEG blocking artifacts analysis [41], [119], JPEG headers analysis [105], thumbnails [103] and EXIF analysis [85], DCT coefficient analysis [25, 27].

CHAPTER 3

EVIDENCES FROM THE EXPLOITATION OF JPEG COMPRESSION

Due to the remarkable number of papers which over the past years covered these topics, the great variety of terms used to define the same components created some confusion. To standardize our exposure we provide a list containing the various features that will be considered. After discussing the different choices found in literature, in some cases we will propose our personal one, to be used in the rest of the paper:

- the terms “quantization" and “compression" lead to some misunderstanding in their interpretation. The core of JPEG algorithm is the reduction of size of the image file. This step is often called *compression*, with a clear figurative meaning, and is obtained with a mathematical operation called *quantization* which consists in dividing (and rounding) DCT coefficients for specific integer values. These divisors are coefficients of an 8×8 matrix, called *quantization matrix*, and are referred as *quantization step* or *compression step*. For these motivations, the two terms are often used indifferently. In this paper we adopt “compression" to indicate abstract concept, such as the kind of algorithm or its property, and “quantization" to refer to concrete numbers, so we will use “quantization step", “quantization matrix", and so forth.

- In the world of photography, the “noise" is something that affects the image quality. As an example, the one known as Picture Response Non Uniformity (PRNU) [98] is inducted by imperfections on the sensor of the camera. Some other kinds of noise, which are of great importance for Image Forensics, are introduced during the quantization/dequantization phases of the JPEG algorithm, and since they derive from mathematical rounding operations are frequently named as “errors".

- The way to indicate the quantization step is generally yet uniformed. The authors often

refer to it as q when it does not matter neither how many times the JPEG algorithm is applied, nor which is its position in the quantization matrix. If the number of JPEG compression matters, the quantization step of the *n-th* compression is denoted as q_n, or q^n. Instead, if the exact position (i,j) in the quantization matrix of a generic quantization step is important, the term mostly used is $q_{i,j}$, or $q^{i,j}$, with or without parenthesis. Finally, if the level of detail is so high that both the number of JPEG compression and the position of the quantization step inside the quantization matrix must be defined, in literature we can find indifferently $q^n_{i,j}$, or $q_n^{i,j}$, again with or without parenthesis. We will denote the three conditions exposed above, respectively, $q, q_n, q^{i,j}$ and $q_n^{i,j}$.

- The whole quantization table referred to the *i-th* compression is indicated as Q_i.

- A gray-scale image before the JPEG compression can be thought as a 2-D matrix (in case of color images the approach is just a little bit complex), in which the numerical value of every single element is the luminance of the corresponding pixel. As we will illustrate in the next section, during the compression step the JPEG algorithm maps every element of the 2-D matrix from the spatial to the frequency domain, whereas in the decompression phase the values are casted back again in the spatial domain. Every step of the algorithm (during both the coding and the decoding phase) has its own characteristics, and the statistical distribution of the elements which characterizes a particular phase (that will be discussed in deep in Sect.3.3) needs to be examined separately. For this reason, the name of the terms in a same position varies with the step in which they are discussed (during the compression, the decompression, first or after the DCT), and every author made his own choices in this regard. Consequently, the terms and the related features are denoted in many different ways. Avoiding listing all the solutions adopted by the various authors through the years, we list the one we choosed:

  - $x^{(n)}_{i,j}$ indicates a single element in the spatial domain in position (i,j) of a certain 8×8 image block of an uncompressed image yet subjected to n JPEG compressions ($n = 0$ means that the image has never been compressed, $n = 1$ means that it has been compressed and decompressed, $n = 2$ means that the compression/decompression step has been applied for two times, and so on...);

  - $\tilde{x}^{(n)}_{i,j}$ indicates a term (in the spatial domain of an image that has been compressed n times) during the decompression phase, just after dequantization and IDCT steps mentioned in the following Sect. 3.1;

  - $y^{(n)}_{i,j}$ indicates a single element in the frequency domain in position (i,j) of a certain 8×8 image block of an uncompressed image yet subjected to n JPEG compressions, just after the DCT transform ($n = 0$ means that the image has never been compressed);

  - $y^{(n)}_{(q)i,j}$ indicates a single element in the frequency domain in position (i,j) of a certain 8×8 image block of an uncompressed image yet subjected to n JPEG compressions, just after the quantization ($n = 0$ means that the image has never been compressed);
  - $j^{(n)}_{i,j}$ indicates a single element in the frequency domain in position (i,j) of the n-times compressed JPEG image.

- With regard to the distributions of the terms in the various steps of the JPEG algorithm (both in the coding and in the decoding pipeline), we will use:
  - d_n is the distribution of the terms just after the DCT transform during the *n-th* encoding phase, and $p_n(x)$ its probability density (PDF) function;
  - $d^{'}_n$ is the distribution of the dequantized DCT terms just before the IDCT transform during the *n-th* decoding phase, and $p^{'}_n(x)$ its PDF function;
- the rounding function is indicated either with the word *round* or with the square brackets $[\ldots]$;
- ϵ_q is the quantization error;
- ϵ_r is the rounding error;
- ϵ_t is the truncation error.

3.1 JPEG Compression Engine and DCT Transform

Starting from a RAW color image, the main steps of JPEG algorithm are the following: the luminance component (also indicated as channel) is separated from the chrominance ones converting the input image from the RGB to YC_bC_r space. Later the two chroma channels are subsampled, every image referred to a channel is partitioned into 8×8 non-overlapping blocks and their values are converted from unsigned integer in the range [0,255] to signed values belonging to the range [-128,127]. At this point a DCT transform is applied to each one of them, followed by a *deadzone quantization*, using for each DCT coefficient of the 8×8 non-overlapping block a corresponding integer value [1]. This part of the algorithm is responsible, at the same time, for the powerful compression obtained by JPEG (because many DCT coefficients are reset) and for the loss of information, since the rounding function is not perfectly reversible. For this reason indeed, the entire procedure is said to be *lossy*. The quantized and rounded coefficients, obtained just rounding the results of the ratio between the original DCT coefficients and the corresponding quantization steps, are then transformed into a data stream by mean of a classic entropy coding whose parameters, together with metadata, are usually inserted in the header of the JPEG file to

[1] From here on indicated generically as q, or q_{ij} to indicate the specific quantization step in position (i,j) belonging to a 8×8 quantization matrix.

allow a proper decoding.
In case of a forgery operation the entire process needs to be inverted, since to visualize the image it must be subjected to a IDCT (inverse DCT) transform, and then dequantized. Once visualized, the image can be manipulated and finally compressed again, most probably with a different quality factor (QF from here on) since it is unlikely that the editing software is set up with the same parameters as the inner software of the camera. The entire pipeline is exposed in Fig.3.1.

3.2 Strengths and Utility of the Discrete Cosine Transform

From the paper by Ahmed et al. that firstly developed the DCT [12], the increasingly rapid growth of image processing algorithms, and in particular of compression methods like JPEG for image or MPEG [144] for video, has led to the success of this transform. Many books [13], tutorials [10], papers [18, 27, 81, 154] defined the mathematical details concerned the DCT, and also its application to solve real problems [73]. This literature has been chosen as a main guideline to briefly summarize its intrinsic details.
From a mathematical point of view, the DCT block transform is a linear and invertible function belonging to the family of the *Discrete Trigonometric Transforms* (DTTS), that can be defined as a finite length mapping from a $L^2(R)$ space to another $L^2(R)$ space, and can be generally defined as:

$$\mathscr{F} : \mathbb{R}^n \longrightarrow \mathbb{R}^n, n \in \mathbb{N}. \tag{3.1}$$

Some of the most important motivations for the use of DCT in image compression are the following:

- **It decorrelates the image data**: normally an image contains areas with uniform or slightly varying brightness, bounded from the so-called *edges* which instead cover a limited area. Therefore, neighboring pixels are strongly correlated. The idea is expressing the image values through a linear combination of coefficients that are (ideally) not correlated. This particular Fourier transform [68] is able to remove redundancy between neighboring pixels. The obtained uncorrelated coefficients can then be independently encoded, thus reducing the total entropy of the image data and allowing a higher compression efficiency.

- **It is an orthogonal transformation**: this arises from the property of the transformation matrix to be composed by orthogonal columns and so it is orthogonal itself [13, 81].

- **Normalization**: for each column vector v_j of any discrete cosine transform kernel matrix holds the following:

$$\|v_j\| = 1, i \in [1, \ldots, 8]. \tag{3.2}$$

- **Orthonormalization**: Since both the columns and the rows of the transformation kernel are orthogonal and normalized, this matrix is said to be *orthonormal*. This results in a significant decrease of the computational complexity, being the matrix inversion reduced to a matrix transpose.

- **Efficient algorithms for its computation are available**: the separability together with the symmetry and the orthogonality allows building a fixed transformation matrix that can be computed separately. Moreover it is not a complex transform, (compared i.e., with the Fourier Transform [13]) so there is no need to encode information about its phase.

3.3 Statistical Distribution of the DCT Coefficients

The first insight on the distribution of DCT coefficients was given in [142] and [131], even if until [152] substantially there were only a lot of proposals and hypotheses not supported by real scientific background. In their paper instead, Reininger and Gibson performed a set of Kolmogorov-Smirnov [99] goodness-of-fit tests to compare the various options: Gaussian, Laplacian, Gamma and Rayleigh distributions. Their results allowed concluding that:

- the distribution of the DC coefficients (the one in position $(0,0)$ in every (8×8) block, which represents the average value of the input sequence in the corresponding block, follows a Gaussian law;

- the distributions of the AC coefficients (the remaining ones) follow a Laplacian law.

For a complete and exhaustive analytical discussion about the topic we refer to [110]. In this paper, Lam and Goodman firstly pointed out that in every 8×8 image block pixel values can reasonably be thought as identically distributed random variables, generally with no (or with a weak) spatial correlation. They also hypothesized that, considering typical images, the variance σ^2 of the Gaussian distribution of the terms through the 8×8 image blocks also varies as a random variable. This last distinction turns out to be the determining factor for the shape of the coefficients distributions, as we will see. Since the central limit theorem [76] states that the weighted summation of identically distributed random variables can be well approximated as having a Gaussian distribution[2] with zero mean and variance proportional to the one of pixels in the block, and using conditional probability, they define:

$$d_n = p_n(y_{i,j}^{(0)}) = \int_0^\infty p_n(y_{i,j}^{(0)}|\sigma^2)p_n(\sigma^2)d(\sigma^2). \tag{3.3}$$

They finally concluded that for these two reasons, the image can be represented with a doubly stochastic model and, since the probability density function (pdf from here on) of σ^2 in case of

[2]Note that the central limit theorem holds even when the image pixels are spatially correlated, as long as the magnitude of correlation is less than one.

natural images has been proved to be exponential with parameter λ, is possible to write the distribution of the coefficient as:

$$p(y_{i,j}^{(0)}) = \frac{\sqrt{2\lambda}}{2} exp\{-\sqrt{2\lambda}|y_{i,j}^{(0)}|\}. \tag{3.4}$$

Recalling that the pdf of a Laplacian distribution is defined as:

$$p(y) = \frac{\mu}{2} exp\{-\mu|y|\}. \tag{3.5}$$

it is straightforward to conclude that the pdf in (3.4) is Laplacian with parameter $\mu = \sqrt{2\lambda}$. As a corollary it is possible to state that, since the constant of proportionality which links the variance of the Gaussian distribution with the one of the block becomes smaller as we move to higher frequency, and since the quantization step typically increases with higher frequencies, producing that the DCT coefficients corresponding to those frequencies are very likely quantized to 0, the amount of DCT coefficients equal (or very near) to 0 increase as the positional index approaches to the high frequencies, i.e., the lower right part of the 8×8 block. Incidentally, this is the reason why in a context of First Quantization Step Estimation it will be almost impossible discover all the terms of the quantization matrix.

As an application of these results, a recent work by Raví et al. [148] analyzes DCT coefficients distribution to discover semantic patterns useful for scene classification. In these particular cases, it is observed that different scene context present differences in the Laplacian scales, and therefore the shape of the various Laplacian distributions can be used as an effective scene context descriptor.

For completeness it is necessary to cite also the papers by Muller [65] and Chang et al. [100], where (respectively) the Generalized Gaussian Model and the Generalized Gamma Model approaches are proposed for better representing the statistical behavior of AC coefficients in case of images (about the DC one there is a general accordance). Even if the exposed results are to some extent remarkable, scientific community defining pdf of DCT terms is yet used to refer to Laplacian distribution, which in any case is a special case of the Generalized Gaussian distribution. In our opinion this is justified from several reasons: the Generalized Gaussian Model, as exposed in [110], is a better model only when the kurtosis has a high value (that is not true for all kinds of images). Moreover, these models require in general more complex expressions and extra computational cost compared with the Laplacian one. Besides, these latter approaches are not based on mathematical analysis and empirical tests, thus having the drawback of a lack of robustness.

Finally, in 2004 Lam [178] pointed out that, in case of text documents, a Gaussian distribution is a more realistic model.

3.4 Sources of Error in the JPEG Algorithm: Definitions and Main Approaches

The JPEG algorithm is classified as *lossy* because the compression process applied to the image is not fully reversible. Although this loss of information can be unpleasant, for Image Forensics it becomes useful for investigative purposes. Indeed, in analogy to the cues left by the criminal on the crime scene, some traces left in an image from JPEG compression algorithm, can be used to get useful information for reconstructing the history of the document. Cues or “evidences" are usually related upon the “difference between the image quality *before* and *after* the JPEG compression".

3.4.1 Quantization Error

This is the main source of error, and arises when a DCT coefficient is divided by the corresponding term of the quantization matrix, and the result is rounded to the nearest integer. It is formally defined as:

$$\epsilon_q = y_{i,j} - y_{(q)i,j} \times q = y_{i,j} - round\left(\frac{y_{i,j}}{q}\right) \times q. \tag{3.6}$$

where $i, j = (0 \dots 7)$, $|\dots|$ is the abs function, q is the quantization step (i.e., the $(i,j)^{th}$ term of the 8×8 quantization matrix), and $y_{i,j}$ is the $(i,j)^{th}$ DCT term of a generic 8×8 image block.
Unlike rounding and truncation errors, whose definitions are globally accepted since in the various papers are always the ones that we show in the next subsection, in case of the quantization error the bibliography has shown different views. The only aspect with a general agreement is that quantization it is a *non-linear* operation and its output can be modeled as a random variable. The formula in (3.6), which represents the difference between the DCT terms *in the same condition (i.e., dequantized)* before and after the quantization step, represent our point of view. In our opinion, it better allows to highlight the joint effect of the quantization + rounding step in terms of information loss. In addition, and as further proof that this point of view is the most agreed, recent papers facing this kind of error [37, 175] made our same choice.

3.4.2 Rounding and Truncation Errors

Both these two sources of error arise after the IDCT, the inverse DCT transform that drives back the values from the frequency to the spatial domain once the image must be visualized. The output values of this operation (i.e., IDCT) must be transformed in format (from double to 8-bit unsigned integer) and range (from a larger range to [0,255]) to be correctly visualized. Despite they are generally discarded during mathematical modeling of the JPEG algorithm, their effects are not negligible. The formal definitions are as follows:

- Rounding Error: The float numbers after the IDCT must be rounded to the nearest integer value to reconstruct the image in the spatial domain. The difference between the values before and after this process is called rounding error, and it is defined as:

$$\epsilon_r = |\tilde{x}_{i,j} - x_{i,j}|. \tag{3.7}$$

  As exposed in [177], it can be considered as a special kind of quantization with $q = 1$. Fan and de Queiroz in [70] model rounding errors as Gaussian distribution with zero-mean around each expected quantized histogram bin.

- Truncation Error: After the rounding step, always with the goal to reconstruct the image data, the values less than 0 need to be *truncated* to 0, whereas the ones larger than 255 are truncated to 255. The difference between the values before and after this cutoff is the truncation error. it is highly dependent on the QFs used in JPEG compression, and on the tested image dataset [64, 170]. More precisely, the smaller the QF (that means high quantization values), the higher the probability that it arises. Since nowadays the QFs are, by default, considerably high, this kind of error generally occurs with very low probability (less than 1%), and can be reasonably discarded during a mathematical modeling.

3.5 Methods for QSE in JPEG Images DCT Domain

Image Forensics analysis is a process that aims to understand the history of a digital image. At first an image is investigated just to assess its originality: this can be done by figuring out if there are traces of previous JPEG compression(s) or by discovering other kind of alterations with methods that do not exploit JPEG compression (these methods will be not covered in this survey). If a previous JPEG compression is detected the image can be stated to be "not original" or altered in someway. Thus further analysis can be carried out to estimate the first Quantization Step and reconstruct the original JPEG compression parameters of the image in order to identify the source camera that acquired the analyzed image, the editing software that edited it or, as most recently debated, the Social Network Service from which it was last downloaded. Figure 3.1 schematizes the Image Forensics analysis pipeline described above. It is clear that the JPEG Double Quantization Detection (DQD) and the Quantization Step Estimation (QSE) are two fundamental steps in the overall process. Despite other possible criteria, we decided to survey DQD and QSE methods by dividing them into categories with names that recall what is exploited from image data. Thus the categories are for methods based on:

- Probability distributions on DCT coefficients;
- Benford's Law;
- Benford's Fourier Coefficients;

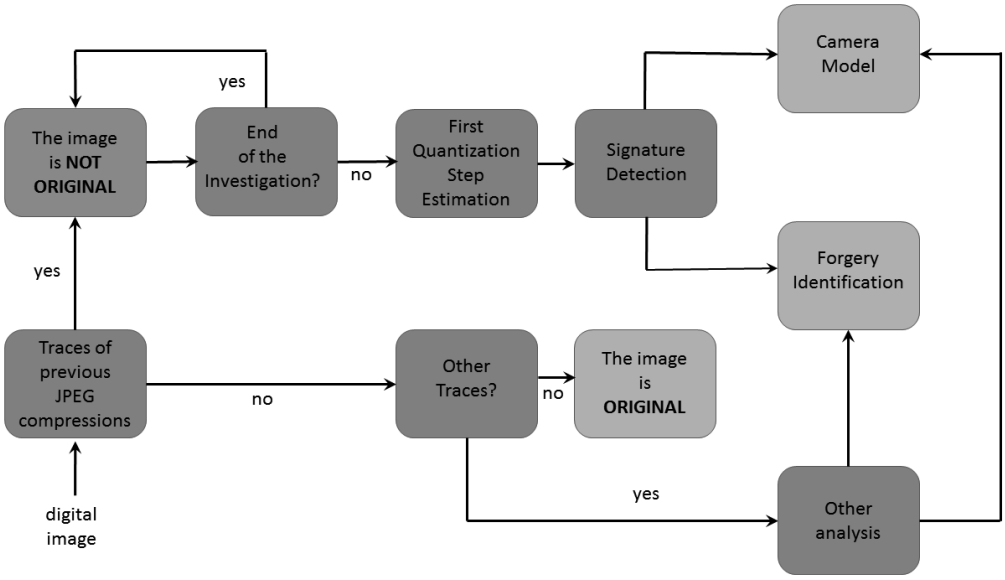


Figure 3.1: When an image is analyzed for forensics purposes, the goal of the examiner can span from the (stand-alone) verification of its originality, up to the investigation of further and deeper details. He might have to ascertain which kind of devices took the image (Source Camera Identification), or/and which areas of the image have been manipulated (Forgery Localization).

- Neural Networks encoding and classification;
- DCT coefficients comparison;
- Histograms and filtering;

3.5.1 Methods based on Probability distributions on DCT coefficients

In [69] and [70], Fan and de Queiroz describe a method to determine if an image in bitmap format has been previously JPEG-compressed and further to estimate the quantization matrix used. They assume that if there is no compression, the pixel differences across 8×8 block boundaries should be similar to those within blocks, while they should be different if the image has been JPEG-compressed. The method first define two functions taking into account inter and intra-block pixel differences. The energy of the difference between histograms derived by these functions is then compared to a proper threshold, and the result allows highlighting the presence of prior compression (also with very high QFs). After detecting the compression signature, they present a method for the maximum likelihood estimation (MLE) of JPEG quantization steps and showed its reliability via experimental results. This work deserves to be quoted mainly because it proposes (for the first time in multimedia forensics) an estimation of the lattice structure in the DCT domain, even if limited on the case of quantization performed for a JPEG compressed image

without downsampling of color components. As a critical remarks, we point out that both methods are limited to QF ≤ 95.

In [36], which in turn took the cue from [74], Bianchi et al. started to face with a particular scenario that they called Single Compression Forgery for JPEG images. This is the situation in which a part of a JPEG image is patched over an uncompressed image (copy-paste or cut/past operation) and the result is JPEG compressed. The core of the method is the use of Bayesian inference to assign to each DCT coefficient a probability of being double quantized, giving the possibility to build a probability map that for every part of the image tells if it is original or tampered. One of the parameters needed to calculate the probability is the quantization step of the first compression that, in this case, is iteratively estimated.
In [35], which is itself a refinement of [34], the authors build a likelihood map to find the regions that have undergone to a double JPEG compression. In particular they observe that the distribution of the DCT coefficients of a tampered image, considering the above scenario, can be modeled as:

$$p(x;q_1,\alpha) = \alpha \cdot p(x|H_0) + (1-\alpha) \cdot p(x|H_1;q_1). \tag{3.8}$$

where α indicates how strong is the probability that the DCT coefficient has been single quantized (hypothesis H_0, $\Rightarrow$ it belong to a non-tampered part), or double quantized (hypothesis H_1, $\Rightarrow$ it belong to a tampered part). As can be seen, among the parameters required to correctly identify this likelihood map and modeling the double compressed regions, q_1 (the quantization step of the primary compression) is crucial. The authors estimate q_1 using the EM algorithm over a set of candidates. This procedure is replicated for each of the 64 DCT coefficients that composes the first-compression matrix. Besides its results, the method is important because it takes into account two kinds of traces left by tampering in double-compressed JPEG images: aligned and non-aligned, something that was considered, to the best of our knowledge, only few times before [20, 179] and never after. These two scenarios, respectively referred as A-DJPG and NA-JPG, arises depending if the DCT grid of the portion of image pasted in a splicing or cloning operation is (or not) aligned with the one of the original image. Also in these papers is underlined the difficulty to correctly estimate q_1 when it is $\leq q_2$. Indeed, their results are heavily affected from this problem, as they pointed out in their conclusions.

In [175], mainly dedicated to tampering detection by means of a proper comparison of the different distributions of DCT coefficients between tampered and non tampered regions, Wang et al. built up a mathematical model in which the knowledge of q_1 is required. For this reason, even indirectly, it provides a way to determinate the first quantization step in double compressed JPEG images. Indeed, starting from the expression of a generic double compressed DCT term without taking into consideration truncation and rounding errors, the authors model the probability distribution of the absolute values of the DCT coefficients in a tampered image. From this statistical model they define a likelihood function and use the EM algorithm for the estimation of the unknown parameters. In doing so, they also take into account the truncation and rounding

errors, referring to the work of Bianchi et al. [36]. We want to point out that also in this work the case $q_1 < q_2$ gives unsatisfactory results.

3.5.2 Methods based on Benford's Law

In [80], the first digit law ([95]) is applied to estimate the JPEG-compression history for images in bitmap format, by means of a Support Vector Machine (SVM) based classifier. In particular, the provided results include the detection of JPEG single compressions and the estimation of JPEG compression factors even if limited to QFs ≤ 99. The authors demonstrated that the probability distribution of the first digit of the DCT coefficients in original JPEG images (single-compressed) follows this Benford-like logarithmic law:

$$p(x) = N log_{10}\left(1 + \frac{1}{s + x^q}\right). \tag{3.9}$$

with $x = 1, 2, \ldots, 9$, N is a normalization factor, and s, q are the model parameters.
Consequently, they proposed a generalized form of the Benford's law to precisely describe the distributions of the original JPEG images with different Q-factors. Since the first digit of the DCT coefficients in double JPEG-compressed images does not follow the above mentioned law (in the paper is observed that the fitting provided by the generalized Benford's law is decreasingly accurate with the number of compression steps), its presence (or absence) can be used as a signature in a double JPEG processing detection algorithm.
Like all the milestone papers, also this work has been deeply explored over the years. As an example, in [112] is discussed how the performances of this approach can be improved by examining the first digit distribution of each sub band of DCT coefficients independently, rather than analyzing the entire set at once. Again, Feng et al. in [75] outperformed the results given in the above paper with a multi-feature detection method using both linear and non-linear classifications, and in 2012 Li et al. in [114] proposed an approach that uses the Benford's law to detect tampered areas in a JEPG image. One year after, Hou et al. extended the above results by including also *zero* in the set of possible digits for the statistics of the first digit distribution [96]. In the same year the authors of [125] exposed an Antiforensics method to fool the statistics connected to the Benford's law in case of the detection of double compressed images. Another interesting question is answered by the application of this statistical rule in [126], where Milani et at. exposed a method able to detect how many JPEG compressions have been applied to the image.

3.5.3 Methods based on Benford's Fourier Coefficients

At the same time, in a work by Pasquini et al. [137] is proposed a binary decision test, based upon the Benford-Fourier theory, to distinguish the images that was previously JPEG compressed starting from images stored in an uncompressed format. Always leveraging the same theory,

Pasquini et al. in [136] faced also multi (up to three) JPEG compressions , exposing a method for its identification starting from JPEG images. Their approach is also extended to the estimation of the quantization steps, passing through a training phase followed by a testing one, obtaining good results compared with [35], also for the challenging case of $q_1 < q_2$, even if limited to an aligned image grid.

3.5.4 Methods based on Neural Networks encoding and classification

In [118], which is devoted to retrieve the first quantization step in double compressed JPEG images, J.Lukáš and J.Fridrich point out for the first time that the problem needs to be addressed separately, depending upon the relations between the two quantization steps:

- $q_1 = q_2$ or q_1 is a multiple of q_2;
- $q_1 > q_2$;
- $q_1 < q_2$ and q_1 is not a multiple of q_2;

In particular this paper, which also has the goal to detect if a given JPEG image has been previously double compressed (*Compression Detection* problem), provides a motivation for the choice to study separately every position in the 8×8 image block (since DCT is an orthogonal transform, changes in one coefficient should not affect other ones), and for considering only DCT coefficients of the low frequency part of the 8×8 image block (approaching to the 64^{th} coefficient, following the same zig-zag order of the JPEG entropy coding, the amount of information that is possible to extract becomes progressively smaller). Nevertheless, the main limitation of the work lies just in the fact that it only takes under consideration DCT coefficients in the first three positions of the block, that is not enough helpful to uniquely identify a quantization matrix.

3.5.5 Methods based on DCT coefficients comparison

Although the paper [91] was not strictly dedicated to recover the quantization steps of the first quantization matrix, but the tampered regions (the so called “ghosts") in a double compressed JPEG image, as a part of the proposed algorithm the author suggested a method to estimate q_1 . In particular he proposes to carry out a third quantization, then computing the error between the DCT coefficients before and after this step. By varying the QF of the third compression, the error function produces two minima in correspondence of q_1 and q_2. Actually, for a particular set of cases (i.e., both coefficients are prime numbers), the method obtains interesting results and q_1 can be easily found since it corresponds to an evident local minima of the error function. However, as exposed in [83], it can be noted that in the majority of cases the original quantization step cannot be easily inferred. Nevertheless, in 2012, Zach et al. [183] present a method that fully automates the detection of JPEG ghosts.

In [170], W. Luo, J. Huang and G. Qiu expose a method to identify Bitmap images that have been previously JPEG compressed (when QF is ≤ 98), assessing quantization steps and detecting quantization tables. An interesting result presented here is that the de-quantized coefficients of a JPEG compression will be well preserved with the highest probability after JPEG recompression, also when compared with many others including the quantization table of ones ($QF_2 = 100$), when the second compression has been exploited using the same quantization table of the first one ($Q_2 = Q_1$). The strength of their approach lies in the fact that it works also when the test image does not have enough statistics, i.e., the tampered region within an image is just a small patch such as a face, some numbers on a plate, and so on. Indeed the reported outcomes continue to be significant also when the size of the images are reduced to a block of 64×64 pixels, in case of the detection of the quantization table. For this purpose, first the proposed algorithm JPEG recompresses the image $X^{(1)}$ (a singly-compressed JPEG image) with all the possible quantization tables belonging to a known set with QFs are ranging from 1 to 100, obtaining 100 different decompressed version of $X^{(2)}$. Once obtained the QF, the compression matrix is uniquely identified.
One big limitation of this paper is that it assumes that the primary quantization matrix belongs to a set of known matrices, whereas in a real scenario this is not generally true, since today image acquisition devices often create on site their own best quantization tables.

In their paper [64], F. Huang et al. cope with the detection of double JPEG compression when the primary and secondary quantization matrix are the same. In this scenario, the “characteristic framework" introduced by different quantization matrices is missing, causing the usual detection methods to not work properly.
They start observing that, caused by the three kinds of errors which arise in the JPEG compression/decompression pipeline (quantization error, rounding error and truncation error), the number of different DCT coefficients in case of multiple compressions (recompressing a JPEG image over and over again) with the same quantization matrix, decreases as the number of compressions rises. To this aim, they define as C_n the rate of coefficient change between two subsequent JPEG quantizations, which is dependent on the number of nonzero DCT coefficients and the different DCT coefficients between the quantization. Subsequently, they take about 4000 images of roughly equal size from three different datasets and double compress them with the same quantization table. In this way they can observe that, in case of single and double compression, the average value of C_1 is heavily depending on the chosen database.
The above observation gives us the opportunity to introduce the important topic consisting on the need of a shared and agreed set of images for the validation tests in every proposed method. At the present time the scientific community does not always take it under the necessary consideration, although it is a key component in order to judge the robustness of a method. In Section 3.6 we will add further considerations about this aspect.
The algorithm proposed in this paper starts recompressing the examined JPEG image J with the

same compression matrix obtaining $J^{'}$. Then it calculates the number D of different coefficients between J and $J^{'}$. The next is the key step, since expects to entropy decoding $J^{'}$, randomly selecting and modifying a “proper" amount of its DCT coefficients and finally decoding the whole image again obtaining $J^{'}_{m}$. After that, decompressing and recompressing $J^{'}_{m}$ with the same quantization table to get $J^{''}_{m}$ and set D^{1}_{m} as the number of different DCT coefficients between these two images. The last two steps are repeated a certain number of times and an average value $\bar{D}_{m}$ is calculated. At this point the proposed rule for discriminating between singly and doubly compressed images is:

$$\begin{cases} \bar{D}_{m} \geq D & \Longrightarrow \text{J is double compressed.} \\ \bar{D}_{m} < D & \Longrightarrow \text{J is singly compressed.} \end{cases} \tag{3.10}$$

Further considerations regarding the statistical property of the DCT terms, compared between subsequent compressions, allow the authors to state that not only as the number of compressions increases the amount of different DCT coefficients decreases, but the same happens for the difference between two subsequent D_{i} (i.e., $D_{n} - D_{n+1} > D_{n+1} - D_{n+2}$). Since the correct choice of the amount of DCT coefficients to modify is the key for the success of the method, a big part of the paper is devoted to the selection of this term. The results are quite significant: if the QF is no less than 90, the final detection accuracy rates are constantly higher than 90 percent for all the image dataset used. The authors also extended their approach to discriminate between single-triple or double-triple compressions, and declare that, assuming to find a suitable “proper" ratio their method can detect four times and further JPEG compressions.
In the following years, the same problem has been addressed in [121, 156], whereas more recently Lui et al. in [116] proposed a method to detect the presence of forgery in images where the modified parts have been generated with the same quantization matrix of the background.

3.5.6 Methods based on Histograms and filtering

The paper [140] it is an improved version of the method (based on NN) developed in [118], where the histogram is also used to train an SVM. More precisely, in this approach the double compression detector is based on a Support Vector machine (SVM) with Gaussian kernel, and the training is obtained with a collection of 9 DCT coefficients derived from the first order statistics of individual DCT modes of low frequency DCT coefficients. The algorithm not only detects double compressed images, but also can be used as a detector of the Primary Quantization Step, and for this purpose, a set of SVM-based multiclassifiers (a collection of binary classifiers) is used. This approach however, similarly to the work of [75], has been tested only for $QF \in (75, 80)$, since these are the default compression settings of two famous steganalisys algorithms: F5 [176] and OutGuess [132]. A third outcome exposed in the paper is the comparison between the retrieved DCT coefficients and a set of whole quantization matrix, with the purpose of trying to find the “closest" (depending upon a proper metric) quantization table. Moreover, the authors face the

Double Compression detection problem comparing their own feature set with the one proposed in [80], training the classifiers on exactly the same database of images.

In [181], the authors extract a statistic feature called factor histogram from the analysis of the procedure of double quantization. This characteristic describes the distribution of the factors being related to quantized DCT coefficients, and is used to detect double quantization and to estimate the primary quantization matrix in double compressed JPEG images. According to the authors, to derive the notion of factor histogram it is necessary to start with a formula that represents the generic DCT term after the second quantization, taking into account all the kinds of errors. Following considerations about the rounding function, and neglecting the rounding/truncation errors, the authors state that the product $y^{(1)}_{i,j} \cdot q^{(1)}_{i,j}$, (where $y^{(n)}_{i,j}$ indicates a single element in the frequency domain in position (i,j) of the 8×8 image block of an uncompressed image yet subjected to n JPEG compressions, just after the DCT transform, and $q^{(n)}_{i,j}$ is the coefficient in position (i,j) of the quantization matrix used in the $n-th$ compression) belongs to the set of $q^{(2)}_{i,j}$ consecutive integers. At this point they factorize each integer belonging to this set, and collect all of the positive integer factors to form the factor set that is a constraint for the value range of the step size $q^{(2)}_{i,j}$. The next step lies in defining the factor histogram h_f as the histogram of the factor sequence given by a set of k different factor sets obtained as state above by varying $y^{(2)}_{i,j}$ in a nonzero coefficient sequence of length k, in which each component has been double quantized with step size q_1 and q_2. Finally, the authors set up a kind of score assessment strategy associated with this factor sequence. Inside of this rank, for a double quantized sequence with $q_1 > q_2$ the factor histogram will achieve its maximum at q_1, thus allowing to detect the first compression step.
It is important to point out that this approach starts from two hypotheses: the first one, which allows to assert that $q^{(2)}_{i,j} \in F(y^{(2)}_{i,j}, q^{(2)}_{i,j})$, is that $q^{(1)}_{i,j} > q^{(2)}_{i,j}$. This means that the QF of the first quantization is assumed lower than the second one. This assumption also characterizes other approaches, as we will see, indeed the study of the case when $q^{(1)}_{i,j} < q^{(2)}_{i,j}$ is an open challenge for the scientific community also at the present time. The second conjecture, assumes the knowledge of the set of quantization matrices used for the first compression, that authors called QMS (Quantization Matrix System) and represents the JPEG quantization matrices provided by a company (producing either cameras or photo editing software) inside its software. Even if the latter assumption can in some cases correspond to a real scenario, since this information can be discovered by other ways, nevertheless for a fair comparison to the state-of-the-art we think that it would be better to avoid any kind of hypothesis about such information.
Recently, in [180], almost the same group of authors exposed another method based on the statistics of the factor histogram for estimating the JPEG compression history of bitmap images. In particular, the authors move from the observation that this feature decreases with the increase of its bin index for uncompressed bitmaps, whereas exhibits a local maximum at the bin index corresponding to the quantization step for JPEG decompressed bitmaps.

In [84], which refines what has been proposed by the authors in [182], Galvan et al. focus on the determination of first quantization step in double compressed JPEG images, assuming that $q_1 > q_2$. The main novelties of the proposed approach, which refines what has been proposed by the authors in [83], are related to the filtering strategy adopted to reduce the amount of noise in the input data (DCT histograms), and on the design of a novel function with a satisfactory q_1-localization property. Specifically, a two steps filtering has been introduced to deal with two kinds of errors, that the authors called “split noise" and “residual noise". The first one appears in presence of specific conditions related to the first and second quantization steps, when two consecutive multiples of q_2 are related to a generic multiple of q_1. Once the split filter has been performed, a filtering strategy based on the preservation of the monotonicity of the DCT coefficient distribution is employed to remove further impurities. After this filtering, a function with the aim to properly localize q_1 has been designed and evaluated, generating a set of outputs among which a limited number of first quantization candidates are selected exploiting the q_1 localization property of the proposed error function. Finally, the double quantization process is simulated to consider the candidates provided by the previous steps and the best one considering directly histogram values is selected exploiting information coming from the original double compressed image.

3.6 Summary and Discussion on JPEG based Image Forensics Methods

Somehow recalling what exposed in the final part of the Section **??**, we observe that the papers exposed could be divided in three sub-categories according to different criteria. The former criterion is simply the format of the image under examination (an uncompressed image previously JPEG compressed, or a double compressed JPEG image), the second way is related to the peculiar approach used to tackle the problem (SVM, Neural Network, probability theory, comparison between DCT histograms, etc), and finally, a third partition could be done considering the goal that the paper pursued (finding tampered areas, understand if an image has been previously JPEG compressed, or just a stand alone method to retrieve the quantization matrix). To improve the reliability and usability of these methods, together with new ideas a certified set of image datasets also strongly needs to be agreed by the entire Image Forensics community. For every given set of images is important to define the source (indoor or outdoor), the QFs used for the compressions and the dimensions of every element. The list of open problems in this field includes the case of double JPEG compressions with $q_1 < q_2$, and the “non-aligned grid" scenario described in [35], which characterizes the most part of the real forgeries, are far to be completely examined. In addition, a better understanding of the three kinds of errors which heavily affect the image during the compression/decompression/recompression steps in a forgery pipeline, would surely better clarify the overall understanding of the problem.

Recently, the overall feeling for the importance of the issues connected with the Integrity Verification in Multimedia is constantly increasing. The scientific European community has accepted the challenge and has joined its forces around these four major projects:

- Rewind [8]. It is finalized at developing approaches in three main directions of research: the use of Watermarking as a tool for the validation of the authenticity on an image, and revealing traces of resampling and copy-move operations as a means of image forgeries.
- S-FIVE [9]. Dedicated to face problems related to standardization in Forensic Image and Video Enhancement, it has its main focus on the developing of techniques for improving the quality of surveillance video data and other kinds of images.
- Reveal [7]. The project has the aim in developing tools and services that aid in Social Media verification from a journalistic and enterprise perspective.
- Maven [5]. The project addresses the issue of the efficient management of large amounts of multimedia files and of the extreme volatility of every digital asset, by using some of the latest technologies, powering integrity and authenticity verification tools. Its goals ranges from face detection and recognition to image source verification.

3.7 Boundary issues

So far, after some theoretical background we reported the main aspects involved in the topic of this survey, namely the impact of the various kinds of errors arising in a JPEG compression/decompression pipeline, and the analysis of the papers that gave the stronger feedback for this area of Image Forensics. Nevertheless, this work would not be complete without having touched some aspects that, although somewhat minor, could have a strong impact on the robustness of a method and on the reliability of the proposed outputs.

3.8 The Choice of the Right Dataset

The need of standardization already exposed in Sect. **??** returns when we take under consideration the databases used in the tests exposed in the papers listed in the previous sections. In almost all works, the authors referred about the dataset(s) used to check the robustness of their methods. The most frequently cited is the UCID (Uncompressed Color Image Dataset) [158], but also SUN Database [57], Dresden Image Dataset [87], BOSS Image Database [133], NRCS Photo Gallery [6], Kodak Dataset [4], CASIA Tampered Image Detection Evaluation Database [2] and BOWS-2 [1] are considered. This plethora of source is a big obstacle to a correct and fair comparison between the different approaches. Besides, whereas in some works the authors use two or more different sets of images to control the robustness of the proposed approach (i.e., two in [84] and [94], even five in [170]) others tested their method with only one image ([14, 121]),

created their own dataset ([36, 77, 179]), or did not indicate from where the images were taken ([35]). So is clear that there is not only the need of a shared dataset, but is also necessary to answer to other questions, i.e., which is the minimum amount of images required to give an adequate level of robustness to a proposed method, in case of a Neural Network approach with a training step ? And what about their variability in terms of shooting conditions (inside or outside)? But more than this, in case of an approach devoted to detect image forgeries, we may ask for a dataset of forged images yet including different resolutions and a set of different kinds of patches pasted over the original image, and so on. Moreover, in the case of QSE a proper combination of subsequent quantization parameters ($q_1, q_2, \dots$) should be properly assessed, also considering the real scenario where the involved parameters are not well known [**?**]. In the paper by Torralba et al. [165] we found good suggestions about how to create a fair dataset and in general about the fairly use of some existing sets of images. About the first point as an example is suggested to collect the images automatically, instead of in a manual and supervised way. For the latter point is argued that to detect potential sources of bias (or at least to find out the main problematic issues quickly and early, not years after the dataset has been released) it would be better to run any brand-new dataset on a given battery of tests. Finally, in using one dataset with respect to another, authors must consider that sometimes different datasets are used for different applications. For example, due to their characteristics, Dresden Image Dataset is extensively used for PRNU extraction, while UCID can not be used for this purpose.
Recently, the problem of a common dataset where to fairly compare the various approaches is exploited by the RAISE dataset [164]. It is composed by a collection of 8156 high-resolution RAW downloadable unprocessed images, taken from seven different scenarios. Time will tell if this purpose will become a landmark.

3.9 Computational Time

As a rule, every new algorithm proposed to the scientific community is evaluated also considering its computational time. In case of Image Forensics instead, this is not a point taken under consideration. The reason of this lies in the use that a forensic expert will do of the method, in particular the fact that he acts not on the crime scene but after the event. In general indeed any investigation is divided in two stages. The first one (where the image forensics expert rarely is one of the crew) includes the moments in which the crime is accomplished and the police, if present, must take action to prevent it from being led to more serious consequences, or to collect evidences. The second phase starts when the crime is discovered and denounced, after which the law begins to take its course. In this moment, the scientific approach for the correct readability of the investigative cues gives to the expert the possibility to answer to the various questions that arise upon a visual document. For example: is this image original? Has it been tampered? In case of positive answer, where and how? Has this image been taken from that device? and so on.

These questions, that generally regard few images, are transmitted to the expert and generally the time allowed to him by the public executor amounts to 2/3 weeks.
In this scenario, there is a difference between the time taken to test the methods over an entire dataset (possibly some days), that is what the researcher must do for the validation of his method, and the computational time reported by some papers [175] ($\sim 0,5$ seconds for every image) that took for the application of this method to every image. These considerations let understand that, from the investigative point of view, this topic could be of limited interest.
An exception to what stated above can be done when investigators have the need to examine an huge amount of images as quickly as possible. As an example, in the investigations that followed the Boston bombing in 2013 the FBI's call for help caused law enforcement agencies to have access to thousands of footages uploaded by citizens, which sent in what amounted to 29 terabytes of data [**?**]. In those fundamental moments, a fast algorithm to check the originality of the uploaded images before using them for investigative purposes is undoubtedly desirable.

3.10 Antiforensics

Trying to hide the traces of a malicious action, deleting them or preventing their discovery, is a behavior that has always characterized attackers. Also in Computer Forensics, as its methods come out from the scientific community, we witness to the contemporary flowering of approaches intended to make the first ineffective [139, 147]. Even if this natural phenomenon is somewhat desirable, given that in general it allows the development and growth of any science, it would be preferable if, in cases of forensics approaches, anti-forensics methods would not develop to the same speed as their opponent, but the enormous ease to find the details of any scientific discovery makes this hope very difficult to be satisfied. Also for Image/Video Forensics area, several authors exposed ways to fool the forensics analysis of a visual document, both hiding the traces of a forgery and possibly trying to delete them. In [151] Redi et al. exposed a complete list of possible anti-forensics approaches, followed and supplemented two years later by the work of Piva [11]. From this survey on, restricting our discussion to the DCT domain since there are other areas of Image Forensics which developed their own Antiforensics methods [89, 161, 174], we can cite the work of Fan et al. [66], where after having pointed out the potential vulnerability of the quantization table estimation based detector, the authors proposed a method to fool JPEG forensic detectors based upon optimizing an objective function considering both the anti-forensic terms and a natural image statistical model. In the same year Sreelakshmi et al. [160] proposed an anti-forensic technique based on the removal of the compression fingerprints of JPEG compression (the blocking artifacts and the characteristic behavior of the histograms of the DCT terms), and Comesana et al. [50] exposed an approach to fool the histogram-based Image Forensics methods, and also recently Fan et al. in [67] exposed a method with the purpose to hide the information detectable by histograms-based Image Forensics algorithms.

As the Forensics methods generate the Antiforensics ones, these latter have stimulated the creation and development of a set of approaches, which are known as Counter-Antiforensics. This sub-area of Image Forensics has the aim to discover the information about JPEG compression history after the action of an anti-forensics pipeline, considering both the fingerprints that were not deleted and the ones left by the previous anti forensics attack itself. About the bibliography in this challenging sector we want to remember the ones exposed by Lai et al [109], Li et al. [92], Valenzise et al. [166] and the interesting approach devoted to merge together Image Forensics and Counter-Antiforensics methods given by Fontani et al. in Ref. [79].

CHAPTER 4

SOCIAL MULTIMEDIA IMAGE FORENSICS

Image Forensics traditionally refers to a number of different tasks on digital images aiming at producing evidence on the authenticity and integrity of data (e.g., forgery detection) and on the identification of the acquisition device (camera identification) [141],[162], [26]. Digital images are continuously altered starting from the moment they were acquired. Most of the time, these alterations are made by users with precise malicious intents. Typical tamperings are the removal or the insertion of an object in an image, the cropping of an undesired portion of a picture, or the application of particular filters to modify or mask sensible parts (e.g., faces in pedo-pornographic photos). When the tampering is not clearly visible, the problem of detecting it becomes obviously challenging. To solve the forgery detection task, some approaches stand above the others: a group of them looks at the structure of the file (e.g., JPEG blocking artifacts analysis [41], [120], hash functions [23], JPEG headers analysis [106], thumbnails [104] and EXIF analysis [85], etc.); others try to identify the device that acquired the image by making use of PRNU patterns ([49],[61]), or focus on statistical analysis of the DCT coefficients ([28, 82, 150]). A voting approach has been used for the same purpose in [23]. An important task for Image Forensics is finding the camera device that acquired the image. Some in-depth studies ([71] [107]) showed that it is possible to coarsely solve the camera identification task, using the DCT coefficients as a feature. Hence it is clear the importance of the JPEG pipeline in retrieving information about the history of an image. Nowadays Social Networks allow their users to upload and share large amounts of images: just on *Facebook* about 1 billion images are shared every day. What happens when a picture is shared on a social platform? How does the upload process affect the JPEG elements of the image? A Social Network is yet but another piece of software that alters images for bandwidth, storage and layout reasons. These kind of alterations, specifically scaling and re-compression, have been proved to make state-of-the-art approaches for camera identification less precise and reliable ([88],[155]). Recent studies ([128],[47],[43]) have shown that, although

the platform heavily modifies an image, this processing leaves a sort of fingerprint on the image itself. All those studies focus on the analysis of too few Social Networks and specific unrealistic scenarios making their works not general enough. In order to improve state of the art and to deeply understand how SNSs process images, a dataset of images from different camera devices was collected, under controlled conditions. We selected ten SNSs through which we processed the collected images by mean of an upload and download process. By doing this, a dataset of images has been obtained, in order to identify any alterations on JPEG elements. The main discovery of our study was that alterations observed are platform dependent (server-side) but also related to the application carrying out the upload (client-side). This evidence can be fundamental for investigation purposes to understand not only the provenience of an image, but also if it has been uploaded from a given device (e.g., Android, iOS). All the observed alterations allowed to build an automatic classifier, based on two K-NN classifiers and a decision tree fitted on the built dataset. Starting from an input image, the proposed approach can predict the SNS that processed the image and the client application through which the image has been uploaded. The remainder of this chapter is structured as follows: in Sec. 4.1, we describe how the dataset has been built, which social platforms have been considered and what kind of upload methods have been used; in Sec. 4.2, an in-depth analysis on dataset images is reported in order to find alterations that can be coded into a fingerprint for a SNS processing; in Sec. 5.1, our approach for image ballistics on social image data is presented with the obtained classification results. Finally, conclusions and reasoning about possible future works on the topic are discussed.

4.1 A Dataset of Social Imagery

The alterations introduced on images by SNS can be thought as a unique fingerprint left by the SNS. The aim of our study is to discover those fingerprints by analyzing the behavior of the most popular SNSs that allow image sharing. Hence, 10 platforms have been selected. First of all, *Facebook* (www.facebook.com) and *Google+* (http://plus.google.com) were taken into account as being the two most popular platforms where users can share their statuses and multimedia content to a network of friends. *Twitter* (http://www.twitter.com) and *Tumblr* (http://www.tumblr.com) were considered as being representative of the micro-blogging concept. We included also *Flickr* (https://www.flickr.com) and *Instagram* (https://www.instagram.com) as platforms focused on sharing high quality artistic photos with capabilities of image editing and filtering. *Imgur* (http://www.imgur.com) and *Tinypic* (http://www.tinypic.com) were also taken into consideration even if they are not properly SNSs but are very popular platforms for image sharing: users usually link images hosted on them from forums and web sites all over the Internet. Finally *WhatsApp* and *Telegram* were also selected as being the two most popular mobile messaging platforms that, by allowing users to create chat groups, are another big place for image sharing on the Internet. Specifically, the last two services are often involved in forensic investigations.

To discover how SNSs process images, we collected a set of photos with the camera devices listed in Table 4.1. Images were acquired representing three different types of scenes: outdoor scenes with buildings (artificial environment), outdoor scenes without buildings (natural environment) and indoor scenes. When taking a picture, we captured two versions: a High Quality (HQ) photo at the maximum resolution allowed by the device, and a Low Quality (LQ) photo (see also Table 4.1). Capturing images in this way, a dataset with a good variability in terms of contents and resolutions was obtained. Images collected so far were uploaded to each of the considered platforms with two different methods: with a web browser, and with iOS and Android native apps. No further discrimination is needed for web browsers because we observed that alterations are not browser-dependent. Each download was performed by searching for the image file URL in the HTML code of the page showing the image itself. At the end of this phase 2400 images were properly collected. The second upload method was carried out with iOS and Android native apps of each social platform, except for *Tinypic* and *Imgur* that do not possess an official app in stores. Moreover, the upload has been done by choosing images in two ways: by searching in the gallery for a previously acquired image (images from local gallery) and by acquiring the image with the camera app embedded in the app itself (embedded camera app). After uploading all images as described above, all of them were downloaded through the "URL searching technique" previously described. 320 more images processed through 8 platforms were thus obtained. All uploads were performed with default settings. The overall dataset consists of 2720 images in JPEG format and it is available at the following web address `http://iplab.dmi.unict.it/DigitalForensics/social_image_forensics/`.

4.2 Dataset Analysis

The main objective is to find a fingerprint left by SNSs on JPEG structure elements, after an upload/download process, in order to build a classifier for image ballistics. To achieve this goal, all information contained in the JPEG file specification has been analyzed: image filename, image size, meta-data and JPEG compression information. We observed that each upload/download process through the considered SNSs produces different alterations among the above-mentioned elements that could be taken into account as fingerprints of the process itself. Details of these alterations will be described in the following Subsections.

4.2.1 Image Filename Alterations

The analysis of the filename of an image and the comparison with known patterns during an investigation on storage devices can provide information about the platform from which it could be downloaded and the date when it was uploaded. For this reason, we first evaluated if and how each platform modifies the file name. We observed that all platforms except *Google+* do a rename.

Table 4.1: Devices used to carry out image collection. For each device the corresponding Low Quality (LQ) and High Quality (HQ) resolutions are reported.

Model	**Device Type**	**Low Resolution**	**High Resolution**
Canon Eos 650D	Dedicated device	720x480	5184x3456
QUMOX SJ4000	Dedicated device	640x480	4032x3024
Sony Powershot A2300	Dedicated device	640x480	4608x3456
Samsung Galaxy Note 3 Neo	Android 4 Phone	640x480	3264x2448
HTC Desire 526g	Android 5 Phone	640x480	3264x2448
Huawei G Play Mini	Android 6 Phone	640x480	4208x3120
iPhone 5	iOS 6 Phone	640x480	2448x3264
iPad mini 2	iOS 8 Pad	640x480	800x600

Table 4.2: Renaming scheme for an uploaded image with original filename IMG_2641.jpg. The new file name for each platform is reported (Image IDs are marked in bold).

Social	**Rename (image ID in bold)**	**Image Lookup**	**Other information**
Facebook	11008414_**746657488782610**_8508378989307666639_n.jpg	YES	Upload resolution
Flickr	26742193671_**8a63f10c85**_h.jpg	YES	Download resolution (h=1600)
Tumblr	tumblr_**o3q9ghRCRh1vnf44lo9**_1280.jpg	YES	Download resolution (1280)
Imgur	04 - **Dw0KXG2**.jpg	YES	
Twitter	**CdqCPQ-WAAAzrHI.jpg**	YES	
WhatsApp	IMG-20160314-WA0038.jpg	NO	Receiving Date (2016-03-14)
Tinypic	1zqdirm.jpg	NO	
Instagram	1689555_**169215806798447**_744040439_n.jpg	YES	Upload Resolution
Telegram	422114602_5593965449613038107.jpg	NO	

Table 4.3: Alterations on JPEG files. The EXIF column reports how JPEG meta-data are edited: maintained, modified or deleted. The File Size column reports if a resize is applied and the corresponding conditions. The JPEG compression column reports if a new JPEG compression is carried out and the corresponding conditions (if any).

Social	**EXIF**		**File Size**		**JPEG Compression**	
	Camera Data	**Other Data**	**Resize**	**Resize Condition**	**Re-Compression**	**Re-Compression Condition**
Facebook	Delete	Delete	Yes	LQ: $M > 960$ HQ: $M > 2048$	Yes	Always
Google+	Maintain	Maintain/Edit	Yes	$M > 2048$	Yes	$M > 2048$
Flickr	Delete	Maintain/Edit	Yes	Depends on options	Yes	Depends on options
Tumblr	Maintain	Maintain/Edit	Yes	$M > 1280$	Yes	$M > 1280$
Imgur	Delete	Delete	No	Never	Yes	Image Size (MB) > 5.45 MB
Twitter	Delete	Delete	Yes	$M > 2048$	Yes	Always
whatsApp	Delete	Delete	Yes	$M > 1600$	Yes	Always
Tinypic	Maintain	Maintain/Edit	Yes	$M > 1600$	Yes	$M > 1600$
Instagram	Delete	Delete	Yes	$M > 1080$	Yes	Always
Telegram	Delete	Delete	Yes	$M > 2560$	Yes	Always

As an example, in Table 4.2 the new names for an uploaded file with name "IMG_2641.jpg" are reported. The column "image lookup" describes the presence into the new filename of an ID useful to reconstruct an URL that points to the web location where the image file is stored.

Facebook, *Flickr*, *Tumblr* and *Instagram* use the image ID and the platform public API (e.g., Graph for *Facebook*) to build the corresponding URL. *Twitter* and *Imgur* allow finding the image on the respective platform by navigating to the URL:

- https://pbs.twimg.com/media/<IMAGE ID> for *Twitter*;

- http://imgur.com/<IMAGE ID> for *Imgur*;

The other platforms do not present an image ID.

Moreover there could be also other useful information like the receiving date (for Whatsapp) and the image resolution (*Facebook*, *Flickr*, *Tumblr* and *Instagram*) coded into image filenames.

File naming alone can solve the problem of identifying the SNS, but it is weak to be used in digital investigations, because filenames can be easily modified by the user. For instance on *Instagram*, the easiest way to download an image is by right-clicking it and choose "save as". By doing this, the browser instantly modifies the name of the image that is going to be downloaded.

4.2.2 Image Size Alterations

A stronger evidence than file naming is the resize of the uploaded images on each platform. A fine-grained test was performed by using synthetic images derived from our dataset and resized at different scales.

On most platforms, resizing is applied if and only if the input image matches certain conditions. This condition is linked to the length in pixels of the longest side M of the original image, where $M = max(width, height)$. If M is greater than a threshold, a resizing algorithm is applied and the resulting image has its longest size equal to the threshold. In Table 4.3, such conditions and the corresponding thresholds for each platform are reported. *Tumblr* does not rescale uploaded images, while in *Flickr* the threshold is set by the user. When the images are resized, the longest side will be set to a fixed value that identifies, in some sense, the platform that made the operation (see Table 4.3). Let note that, some of the considered platforms use the same threshold value and it is subject to changes over time (for example, during our experiments, the threshold for *Twitter* changed from 1024 to 2048). If a resizing algorithm is applied the obtained image has its longest side with a value corresponding to the platform that made the operation. Unfortunately some platforms share the same values and this values too can vary through time. For example during our experiments the threshold for twitter changed from 1024 to 2048 pixels.

4.2.3 Meta-data Alterations

The best evidence to obtain information, for investigation purposes, are meta-data embedded in JPEG files. These meta-data are technically known as EXIF and can store information like the device that acquired an image, the date and time of acquisition and also the GPS coordinates. For our purposes, we divided EXIF data into two categories: “camera data” which contains all those key-valued that allow to identifying the device that acquired the image and “other data” for every other EXIF information.

In Table 4.3, the results of the analysis on EXIF data are resumed for each platform. In particular, it is reported if "camera data" and "other data" are deleted, maintained or just edited

throughout the processing. Unfortunately, most of the SNSs delete all meta-data, specifically those related to camera data.

4.2.4 Image JPEG Compression Alterations

The images considered in our dataset are all encoded in JPEG format, both the original versions and the downloaded ones. Thus, an analysis on how the SNS processing affects the JPEG compression has been carried out. We focused on the Discrete Quantization Tables (DQTs) used for JPEG compression (extracted by DJPEG: an open source tool part of libjpeg project [3]).

Considering how platforms affect DQTs, it is possible to divide them into two categories:

- Platforms that always re-compress images (*Facebook*, *Twitter*, *Telegram*, *WhatsApp*, *Instagram*);
- Platforms that re-compress images at a given condition (*Google+*, *Tumblr*, *Tinypic*, *Imgur*).

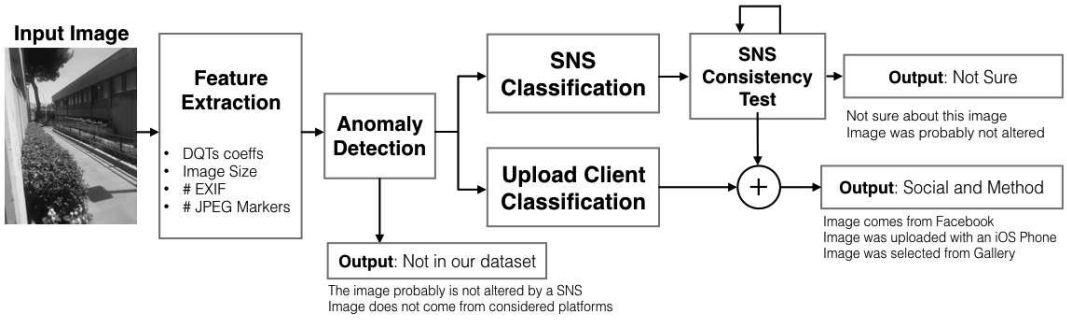
The compression follows the same rules we described for resizing. In fact, a threshold-based evaluation is performed on the longest image side and, if it is bigger than the threshold, the image is compressed using a DQT that will be different from the original one. This is not true for all the considered platforms; *Flickr* allows the user to choose the threshold (if any), while on *Imgur* the threshold is fixed in terms of size in MegaBytes; specifically, if the input image size is greater than 5.45MB, than the re-compression is performed, otherwise nothing happens (see also Table 4.3). In a same way as described for the resizing algorithm, the condition for re-compression is: if the size in pixels of the longest side of input image M is smaller than a threshold, that is typical of the platform, then a new JPEG compression is applied and the obtained image will have different DQTs than the input one, in terms of their coefficients.

Some additional words are needed for *Flickr* and *Imgur*. As regards *Flickr* as described previously for resizing, the condition is set by the user. While for *Imgur*, the condition is based on size in MegaBytes (MB): if the input image size is greater than 5.45MB than the re-compression is performed, otherwise nothing is done. Finally, for *Imgur* for which the condition is based on size in MegaBytes (MB): if the input image size is greater than 5.45MB than the re-compression is performed, otherwise nothing is done.

4.3 Image Ballistics of Social Data

Starting from the results of the analysis reported in previous Sections, regarding the alterations on JPEG elements of processed images, it is possible to assess that such alterations bring pieces of information about the history of the image but they could be insufficient, if considered alone, for investigation purposes. Hence, we encoded all the observed alterations into a set of features to be used as input for an automatic classifier.

Figure 4.1: Classification scheme for Image Ballistics in the era of Social Network Services. The proposed approach encodes JPEG information from an input image into a feature vector. The obtained feature vector is evaluated through an Anomaly Detector that filters out images not processed by a SNS. If the input image is not an anomaly, the feature vector goes through other two classifiers: a SNS Classifier and an Upload Client Classifier. The output of the SNS Classifier is further processed through a SNS Consistency Test that checks if the features of the input image and the predicted SNS are consistent to re-compression and resizing conditions. The final output depends on this last stage: if all features are compatible with the classified SNS then the obtained prediction, joined with upload client prediction, is outputted. Otherwise the consistency test is repeated, for the next most probable predicted SNS, until it is satisfied or it stalls on the same predicted platform. In this case the overall output will be "Not Sure".



For this reason by considering the alterations observed altogether it is possible to identify them as a fingerprint to be used in an automatic classifier. In order to do this we coded the alterations observed into the following numeric features:

- The DQTs coefficients divided in 64 coefficients for the Chrominance table and 64 for the Luminance one, which represent the JPEG compression alterations. These coefficients were investigated separately with PCA and we obtained an explained variance of 99% for the first 32 coefficients of the luminance table and the first 8 coefficient of the chrominance one;
- Image size (width and height in pixels), which brings information about size alterations;
- Number and typology of EXIF data (key-value couples), which describes meta-data alterations (both camera and other data);
- Number of markers in JPEG files as defined in [124].

The listed features were chosen in order to represent each kind of alteration described in previous Sections.

The Quality Factor (QF) was not considered among them, as it was done in [128], for being dependent on DQTs coefficients and thus not bringing any new useful information. In particular, QF does not have a unique method to be computed and this can be be a source of error for classification purposes.

PRNU was not taken into consideration among our features, because, as already mentioned, the heavy processing done on images by SNSs degrades PRNU approaches for camera identification in terms of accuracy [88].

Given the features listed before and the image dataset described so far, a correspondence between features and the SNS has been established. This is particularly true for platforms that always operate a re-compression and heavily alter images. Starting from this correspondence, an automatic classification approach for image ballistics was built. Given an input image, the proposed method allows knowing not only from which platform it comes from, but also which client application was used to upload the image (browser web application, iOS native app or Android native app). Moreover, for images uploaded from iOS and Android native apps, the proposed approach is able to differentiate between images taken from the camera application embedded into the native apps or images chosen through gallery selection. This demonstrates that fingerprints observable on images are left both by SNSs and the client applications carrying out the upload.

4.3.1 Implementing image ballistics: a classification engine

Given a JPEG image I, our objectives are to define:

1. if there is a compatibility between the non-related JPEG elements of I (i.e. filename, EXIF data) and the processing pipeline of SNSs;
2. if there is a compatibility between the JPEG elements of I and the processing pipeline of SNSs;
3. which SNS is compatible with the JPEG elements of the image, with a certain degree of confidence, and what is the uploading source in terms of operating system (OS) and application.

We represent each image I as a 44-dimensional vector

$$\boldsymbol{v} = \{w, h, |E|, m, l_j, c_k\}, \tag{4.1}$$

where

- $w \times h$ is the size in pixels of I;
- $E = \{key, value\}$ is an associative array containing the EXIF metadata, thus $|E|$ is the number of metadata found in the structure of I;
- m is the number of JPEG markers in I;
- $l_j, j = 0, \ldots, 31$ are the first 32 coefficients of the luminance quantization table;

- $c_k, k = 0,\dots,7$ are the first 8 coefficients of the chrominance quantization table.

Moreover, we define $fn(I)$ as the filename of the image I.

At the first stage, we consider $fn(I)$ and E. If there is a matching between $fn(I)$ and the renaming patterns observed in Section 4.2.1, our approach confirms the compatibility between I and the SNS with the matched pattern. Also, E is taken into account, looking for the "Exif.Image.UniqueCameraModel" key. If it is set, then our system returns that value.

Thus, the whole dataset representation is

$$\boldsymbol{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$$

where N is the total number of images. In order to train the SNS and Upload Scenario classifiers, we augment this representation with the corresponding labels. Thus, the final representation for a generic image I_i is

$$\boldsymbol{I}_i = \{\boldsymbol{v}_i, sns_i, uc_i, sm_i\}$$

where sns_i is the SNS, uc_i is the client application and sm_i is the image selection method.

Our classifier performs a two-steps analysis. First, we implement an Anomaly Detector to exclude the images that have not been processed by SNSs, then we run in parallel a K-NN Classifier and a Decision Tree [146] to asses respectively the SNS of origin and the uploading scenario (OS + application).

Given the representations $\boldsymbol{v}_{I_1}$ of an image I_1 and $\boldsymbol{v}_{I_2}$ of an image I_2, we define the cosine distance between $\boldsymbol{v}_{I_1}$ and $\boldsymbol{v}_{I_2}$

$$d(\boldsymbol{v_1}, \boldsymbol{v_2}) = \frac{\boldsymbol{v_1} \cdot \boldsymbol{v_2}}{|\boldsymbol{v_1}||\boldsymbol{v_2}|} \tag{4.2}$$

as a measure of similarity between I_1 and I_2. Therefore, it is possible to build a distance matrix $\boldsymbol{D}$ of size $N \times N$ where the element d_{ij} is equal to the distance between the images I_i and I_j. We will refer to the $r-$th row of this matrix as $\boldsymbol{D}_r$ and to the $c-$th column as $\boldsymbol{D}^c$. It is important to note that $\forall\, I_i, I_j,\ 0 \leq d(\boldsymbol{v_i}, \boldsymbol{v_j}) \leq 1$, and specifically, the more is the similarity, the more the distance will be closer to 1. Exploiting this property, we define the Anomaly Detector as

$$a(\boldsymbol{v}_i, \boldsymbol{D}) = \begin{cases} (\boldsymbol{v}_i, i) & if \sum_{j=1}^{K} d_{ij} < T \\ \text{not processed} & otherwise \end{cases} \tag{4.3}$$

where $T \in [0, K]$ is defined as the Anomaly Threshold. In other words, since the more two images are similar, the more their distance will be closer to 1, we make sure that at least $\lfloor K \rfloor$ samples in our dataset are similar to the query image representation. Then, when $a(\boldsymbol{v}_i, \boldsymbol{D}) = 0$,

the representation is far apart the samples, and we can state that probably the image has not been processed.

The output of a is then used as input by K-NN (4.4) and Decision Tree algorithms [146].

$$knn(\boldsymbol{v}_i, i) = sns_j \mid d_{ij} = \min \boldsymbol{D}^i \tag{4.4}$$

$$dt(\boldsymbol{v}_i, i) = (uc_j, sm_j) \tag{4.5}$$

where uc_j and sm_j are the leaves obtained following the path with $\boldsymbol{v}_i$ as input. Hence, the classification scheme, shown in Figure 4.1, can be formalized as follows

$$C(\boldsymbol{v}_i, \boldsymbol{D}) = knn(a(\boldsymbol{v}_i, \boldsymbol{D})) \oplus dt(a(\boldsymbol{v}_i, \boldsymbol{D})) \tag{4.6}$$

K-NN algorithm looks for the closest sample in the dataset, and assigns the same SNS to the query image. A Decision tree (Eq. 4.5) builds classification in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes. The algorithm used for building the decision tree is the ID3 [146] which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy to construct a decision tree by evaluating $\boldsymbol{v} \in \boldsymbol{V}$.

Finally, the output of the K-NN Classifier sns_j is processed through a SNS Consistency Test. Let be $S = \{sns_1, \ldots, sns_n\}$ the set of SNSs that operates a re-compression at the condition $max(w,h) > C_{sns_i}$ where C_{sns_i} is the conditional threshold for the $i-$th SNS and w and h as listed in Table 4.3.

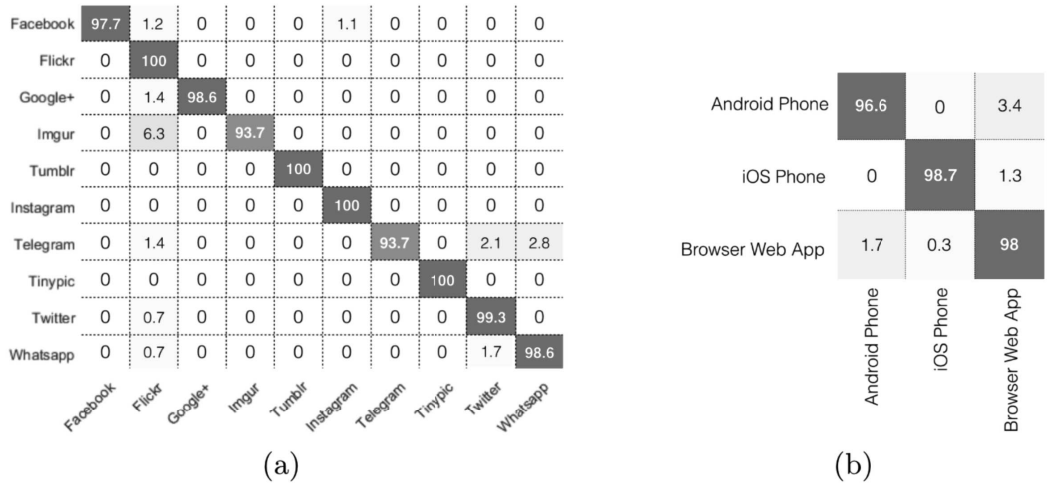
Given that $sns_j \in S$, if $max(w,h) < C_{sns_j}$ it is an anomaly. The test is then repeated for the next most probable prediction from the SNS Classifier until the corresponding condition is satisfied or the loop stalls on the same SNS prediction. In this last case, the result of the classification is "not sure"; otherwise, a SNS prediction is reached and outputted (sns_j) with the predicted upload client application (uc_j) and image selection method (sm_o).

Figure 4.1 shows a schematic representation of the proposed approach.

4.3.2 Classification Results

In this Section, validation results for the proposed approach are reported to demonstrate its goodness. The anomaly detector was validated by taking from our dataset 240 random images that suffered alterations, and 240 images that did not pass through any alteration. The anomaly detector achieved the best error rate, equal to 3.37%, with K = 3 and T = 2.90. The entire approach for image ballistics described in the previous Section was then tested through a 5-fold cross

Figure 4.2: Confusion Matrices obtained from 5-cross validation on our dataset. The reported values, are the average accuracy values (%) in 5 runs of cross validation test. (a) Confusion Matrix for Social platform Classification, (b) Confusion Matrix for upload method classification.



validation test. Best Ks and T were found through grid-search hyper-parameter tuning method. In Figure 4.2, confusion matrices reporting the average value through the 5 runs are shown.

The accuracy obtained for the SNS classification task was 96% with best K equal to 3 while the accuracy value for the upload client classification task was 97.69% with an accuracy of 91% for the prediction of image selection method, given iOS or Android native app as prior.

Different approaches with other classifiers (like linear and non linear SVM) or combination of classifiers (like hierarchical or cascade approaches) were also tested, but the overall results were slightly worse. The classification scheme reported in Figure 4.1 was the best approach we obtained throughout our tests.

In our experiments, we observed that, as happens for different camera devices of the same model [107], different images, from the same platform, have slightly differences in DQT coefficients. This demonstrated the effectiveness of K-NN over other methods for giving to the approach the resilience against little differences while detecting the most-similar SNS fingerprint. We also built a new test set composed of 20 images randomly downloaded from each considered SNS on which we achieved an accuracy in SNS prediction of 94% that is quite similar to the validation results.

Another consideration is needed about the SNSs fingerprints described in this work and regarding the fact that all the alterations observed can change according to software development and releases. For these reasons, the proposed approach is justified for being able to readapt through time, just by updating the reference dataset.

CHAPTER 5

ENHANCING BALLISTICS ANALYSIS EXPLOITING MULTIMEDIA TECHNIQUES

The need to automatically identify information from a digital medium such as an image is a very important issue in many application domains since it would greatly simplify some processes such as, for example, ballistics in fired bullets and weapon identification or the extraction of information from tampered evidences. Nowadays most of investigative processes are manual analysis that take a lot of time, energy and effort and most of times they are subjective and unrepeatable. On the other hand algorithmic analysis can improve results, can make the entire process repeatable and faster. Digital representation of evidences can become the perfect input to algorithms focused on multimedia data analysis.

In this chapter two common investigators' problems are addressed: the identification of the weapon which fired a bullet from the imprinting left on the cartridge of the bullet itself by the weapon; and the reconstruction of serial numbers malevolently canceled on firearms. The techniques proposed are innovative in the way they address the two problems exploiting digital data.

5.1 Weapon identification from cartridge imprintings

5.1.1 Introduction to the problem

In crime scenes in which shooting is involved, the imprintings found on fired bullets and cartridge cases are important evidence. The comparison of the imprintings left by a firearm on cartridges allows forensic examiners to identify a weapon or a weapon model. The final results of this examination can give fundamental evidence useful in court law. Microscopic impritings are impressed, during firing, by the breech face of a pistol slide into the soft metal of the cartridge.

The breech face in contact with the cartridge surface will imprint a negative shape into the cartridge. The shape of the imprinting provide evidence for possible identification of the used firearm. Normally fired cartridge cases are collected in crime scenes and are manually compared by means of comparison cartridges produced by test firearms .

Nowadays LEAs looking for an Automated firearms identification (AFI) system that contribute to shedding light on criminal events by comparison between different pieces of evidence on cartridge cases and bullets and by matching similar ones that were fired from the same firearm. With an AFI the Ballistic evidence can be rapidly analyzed and classified. This can be done exploiting image analysis techniques [113].

The first technique for ballistics evidence analysis was invented at the National Institute of Standards and Technology (NIST), and its called Congruent Matching Cells (CMC) [127]. This method develops the correlation of pairs of small correlation cells instead of the correlation of entire images. The correlation conclusion (matching or non-matching) is defined by whether the number of CMC is high enough. Starting from this method many other have been developed in order to exploit other more refined correlation measure based not only on images but also on 3D topography images. However, most ballistic images stored, in current local and national databases, are still in optical intensity (grayscale) format.

Other studies [102] try to automatically detect the most important information on the cartridge images, namely the firing pin impression, capsule traces, and the intersection of these traces. These traces are compared automatically using the image analysis and identification system to determine the numeric values that indicate the significance of the similarities. These numerical features, that denote the similarities and differences between pistol makes and model.

The motivation for an intelligent firearm identification system is extremely clear. Original brand pistols allow brand-specific features to be formed due to the assembly production technique, which leads to the production of pistols of the same quality. This information is used by forensic experts to intuitively distinguish pistol brands during the evaluation. A feature selection technique could facilitate the ballistic examinations. Moreover state-of-the-art correlation studies on the cartridges and pistol already suggested that there are impressions more effective than others in achieving a better classification. In fact many ballistic evidence search rules compare whole breech face images to make correlations. But not all areas of the breech face and impression are useful for ballistic examination. The detail in these non-related areas may reduce the accuracy of the overall correlation score and rank in a database search.

5.1.2 How to address the comparison issue: shape analysis

Geometric traces formed on the cartridge are specific to each pistol; these traces can be used to facilitate the evidence selection process and provide a logical classification rank to be used for identification of make.

Forensics examiners techniques are based on comparing two microscope images of two

cartridge surfaces fired by the same weapon. As stated before this examination is subjective and looks at finding particular geometric features by means of moving the source of light while analysing the cartridge themselves. This light movement can be seen as a projection from 3D to 2D and describes particular shapes shat examiners want to compare, match and align. A combination of shapes from different projection can be seen as a set of features useful to both compare and align two cartridges. Classifying and aligning shapes could be a trivial task. In the next section a Shape alignment technique will be presented with application to general purposes 2D shape alignment, recognition and retrieval domain. The presented technique will be then applied for the cartridges alignment and classification task in conjunction with Neural Network techniques.

5.1.3 Aligning shapes

There is general consensus that among the cues used by the human visual system, the shape plays a key role for object classification and recognition. The human visual system is one inspiration for investigating computer based shape classification algorithms, although researchers do not attempt to directly emulate it in approach, but merely in the outcome [122]. Motivations beyond that of pure scientific curiosity are provided by several important applications in which the shapes have been demonstrated to be a fundamental feature to solve the task under consideration (e.g., pedestrian detection, road symbols recognition, etc.) [17, 63, 72, 130]. Different techniques have been proposed in literature for both, shape description and classification. A good revision of the main techniques for shape description and classification can be found in [54] and [186]. Approaches for shape representation can be grouped in two different classes [56]: local and global. Local approaches describe shapes as a set of local properties (e.g., the context of each point into the shape [30]), whereas global approaches encode general information about the shape as a whole entity (e.g., the statistical variance of the shape with respect to a prototype shape [169, 172]). Recently, both local and global properties have been combined to obtain better accuracy in shape classification and retrieval [19, 115, 123].

One of the most popular approaches to describe the properties of points of a shape is the Shape Context (SC) [30]. This descriptor has been extensively and successfully used for shape matching and retrieval purposes due to its robustness in finding correspondence between point sets sampled from different shapes. Such technique describes each point sampled from the internal or external contour of an object through a distribution obtained considering the set of vectors originating from that point to all the other sample points of the shape. This description of a point is called Shape Context and it is inherently insensitive to translation and small perturbations of parts of the shape, and can be opportunely normalized to obtain invariance to rotation and scale. Moreover, Shape Context is empirically demonstrated to be robust to occlusions, noise, and outliers. Given sets of shape contexts obtained from different shapes, the matching problem is reduced to a weighted bipartite matching problem solved by employing the Hungarian algorithm [108]. Once

the correspondences between points on two shapes are obtained, a plane transformation (e.g., thin plate spline model) is used to map points from one shape to the other. For classification purposes, a K-nearest neighbor algorithm is used together with a shape distance computed as weighted sum of shape contexts χ^2 distance, image appearance distance, and bending energy (see [30] for all related details).

On the other hand, the Blurred Shape Model (BSM) has been recently introduced in literature as global descriptor for shapes [63]. BSM is able to codify the spatial arrangement of shape parts through a correlogram structure from the center of the shape composed by regions defined with circles and sections. The descriptor is computed by considering a set of points belonging to the contour of a shape. The distances from a point to the centroids of the corresponding correlogram region and from the closest ones are computed and normalized. The inverse of the normalized distances is then added to the corresponding positions of the distribution associated to the involved correlogram regions. This makes the descriptor more robust to deformations. The rotation invariance of the descriptor is obtained by rotating the correlogram taking into account of the region with predominant density. In this last case the descriptor is called Circular Blurred Shape Model (CBSM) [63]. Since the CBSM descriptor encodes a distribution, it is natural to use the χ^2 test statistic as measure coupled with the K-nearest neighbor algorithm for shape classification purposes.

In the last years other new shape descriptors have been presented with features of robustness to affine and non-linear transformation such as [145, 173] or others optimized for the recognition and retrieval task on hand-written sketches such as [42, 62].

In this paper we propose a framework able to combine both, the local properties of Shape Context descriptor (SC) to solve the correspondence problem during the geometric alignment of shapes, as well as the global representation of the aligned shapes obtained through Blurred Shape Model (BSM). First, shapes are described making use of Bags of Shape Contexts to solve correspondence problem between points of two shapes. Starting from the computed correspondences, shapes are aligned by using a voting procedure in the parameter space of the considered model in order to filter out the geometric transformations [24, 143]. Finally, the aligned shapes are described with the Blurred Shape Model descriptor [63] and classified/retrieved by employing the K-nearest neighbor algorithm with χ^2 test statistic. Experiments performed on the seventy-class MPEG-7 dataset, as well as on the seventeen-class symbols dataset used in [63], demonstrate that the proposed strategy outperforms both approached from which our solution originates: Shape Context (SC)[30] and Circular Blurred Shape Model (CBSM)[63]. The proposed approach is also tested on a large scale dataset of hand sketches composed by 20,000 examples distributed over 250 object categories [62]. The performances achieved by our method outperform the current state-of-the-art method [62] on the large hand sketches dataset.

The proof of concept of the proposed approach has been presented in [22]. Compared to [22], we have improved the algorithm in terms of robustness with respect to scale changes. Moreover,

the performance of the proposed approach with respect to perturbations such as rotation, scale and shear has been tested. Finally, several tests and comparisons with state of the art methods, considering the retrieval scenario and a large dataset, have been performed.

The remainder of the paper is organized as follows: Section 5.1.3.1 describes the proposed framework. Section 5.1.3.2 details the experimental settings and reports the obtained results. Finally, conclusions and avenues for further research are given in Section **??**.

5.1.3.1 Proposed Framework

The proposed solution aims to combine the local peculiarities of shape context descriptor (SC) [30] together with the global representation obtained through Blurred Shape Model (BSM) [63]. It is worth noting that a simple combination, in cascade, of Shape Context and Blurred Shape Model does not provide satisfactory results as pointed out by the experiments described in Section 5.1.3.2. Due to the considered model (thin plate spline), Shape Context is able to transform a shape in a very different one. Considering for example a template and a test image belonging to different classes, the alignment procedure employed by SC transforms the test image in a novel image very similar to the target image (see Fig. 5.1). Computing the BSM over the image provided by the SC does not provide hence any useful information. This problem can be mitigated by modifying the standard parameter setting proposed in [30] in order to limit the variability of the thin plate spline transformation. However, also in this case, the overall performance is not satisfactory (see Section 5.1.3.2).

The proposed solution combines hence local (Shape Context) and global (Blurred Shape Model) descriptors by using a robust registration algorithm [24] during shapes alignment. In this way, the original alignment algorithm proposed in [30] is replaced by a different one based on affine transformations to properly combine SC and BSM descriptors.

The overall schema of the proposed classification framework is shown in Fig. 5.2. As first step, shapes are described as a Bags of Shape Contexts taking into account of a pre-computed codebook. The codebook is generated by clustering the set of Shape Contexts [30] computed from points sampled on contours of training shapes. Specifically, the set of centroids obtained with the clustering procedure becomes the codebook used to represent training and testing shapes for the alignment step. From each training/test shape extracted with Canny edge detector on the corresponding image, n points ($n = 100$ in our tests) are uniformly sampled and described with the Shape Context descriptor. Then each point p_i ($i = 1,\dots,n$) of the considered shape, is associated to a tuple with four components: the id label corresponding to the closest centroid belonging to the pre-computed codebook of shape contexts, the direction θ of the tangent at that point, and the point coordinates (x,y). Hence, each training/test shape is represented with a signature $S = \{(id_i, x_i, y_i, \theta_i) | i = 1,\dots,n\}$ which is exploited for alignment purposes.

As second step, the shapes of training set are aligned with respect to the shape to be classified/retrieved. To this aim, we use the registration procedure detailed in [24]. The entries of the

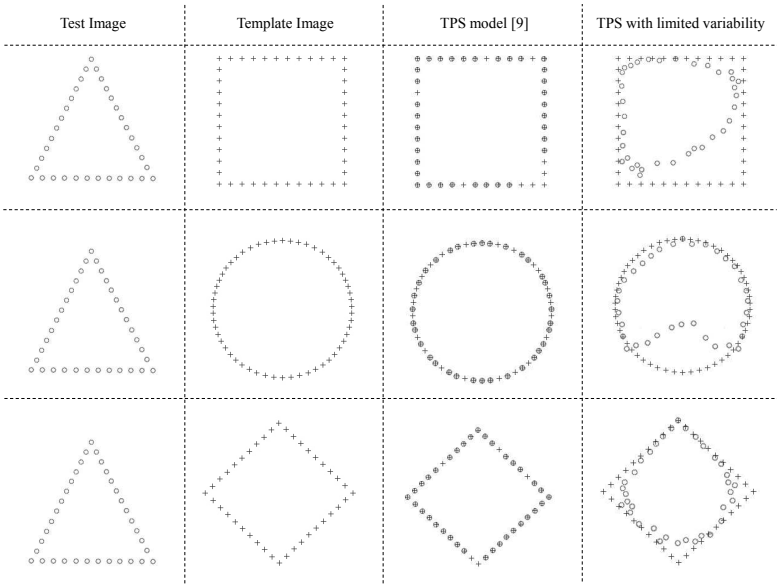


Figure 5.1: Simple combinations of SC [30] and BSM [63]. Shape Context with thin plate spline model transforms a shape in a very different one. Although belonging to different classes, the alignment procedure employed by SC transforms the test image in a novel image too similar to the target. Although the variability of the thin plane spline (TPS) model can be limited by modifying the standard parameter setting used in [30], the overall performance is not satisfactory (see Section 5.1.3.2).

training signatures and the one related to the signature of the shape to be classified/retrieved are matched by considering their id values. To cope with wrong matchings (i.e., outliers), a robust estimator has to be employed. A first filtering is performed considering the differences between tangent orientations (i.e., θ) of matched entries. After this preliminary elimination of outliers, the set of point pairs $((x_{test};y_{test});(x_{training};y_{training}))$, corresponding to matched entries, is aligned by considering a similarity transformation model:

$$x_{training} = x_{test}\sigma\cos\alpha - y_{test}\sigma\sin\alpha + T_x \tag{5.1}$$

$$y_{training} = x_{test}\sigma\sin\alpha + y_{test}\sigma\cos\alpha + T_y \tag{5.2}$$

In order to recover the geometric transformations defined by the parameters (σ, α, T_x, T_y), a robust estimator based on a voting procedure in the parameter space of the similarity transformation model [24] is used. Specifically, each matching votes for a line in the parameter space (Fig. 5.3). By analysing the densest region of this space, a first estimation of the transformation parameters can be obtained. All the matchings voting for the abovementioned solution can be then considered as inliers (see [24] for all mathematical details).

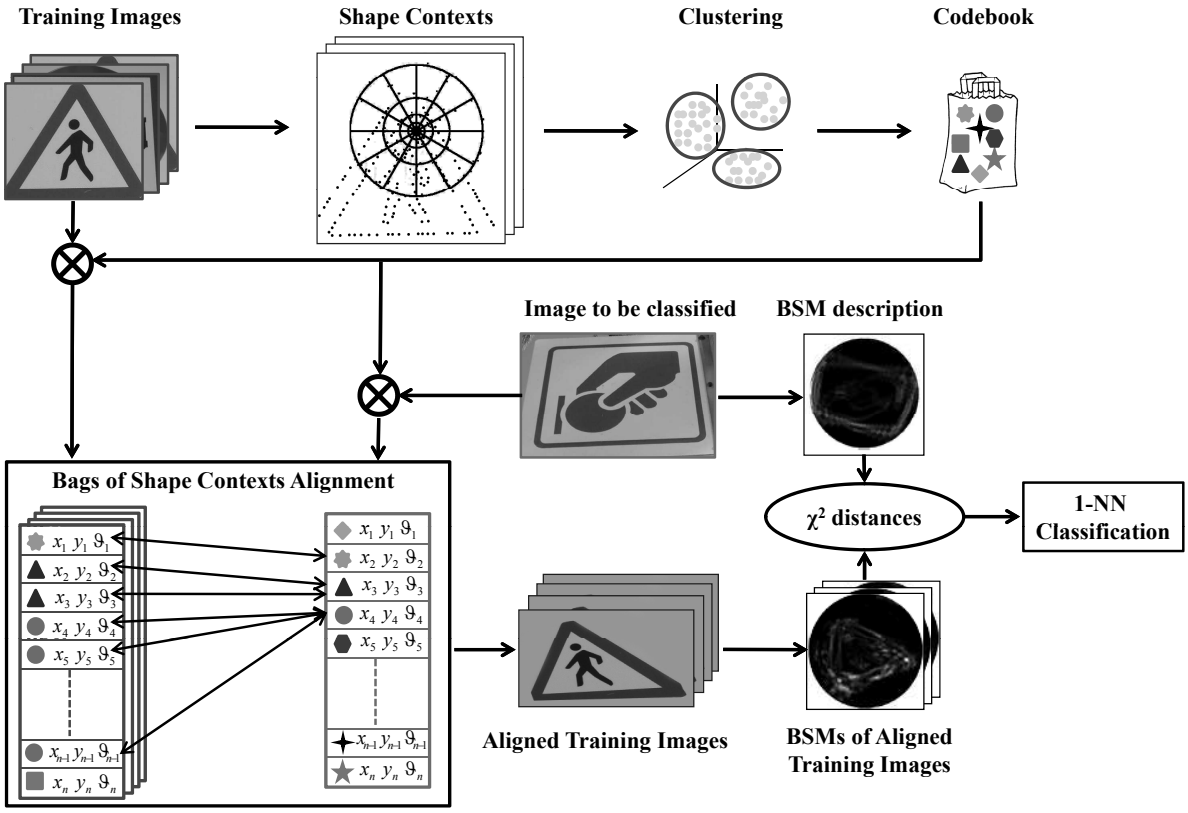


Figure 5.2: Overall schema of the proposed approach. A Bag of Shape Contexts is built and used in combination with BSM to properly classify the input image. A fundamental contribution is provided by the alignment process between Bags of Shape Contexts before exploiting the global BMS descriptor.

In this paper, a further filtering has been performed with respect to the registration strategy proposed in [24] where the outlier rejection (see Fig. 5.3) is performed in a reduced space (α,T_x,T_y). Actually, some matching pairs, close in this reduced space, could be related to very different scale parameter σ. A further filtering is then performed at the end of the algorithm with respect to the scale parameter to discard the remaining wrong matchings.

These matching pairs (inliers) are then used to estimate the parameters of the affine transformation:

$$x'_{training} = ax'_{test} + by'_{test} + T_x \tag{5.3}$$

$$y'_{training} = cx'_{test} + dy'_{test} + T_y \tag{5.4}$$

where (x'_{test}, y'_{test}) and (x'_{test}, y'_{test}) are training and test points considered as inliers and a, b, c, d, T_x, T_y are the six real parameters of the affine transformation. The estimation of the affine

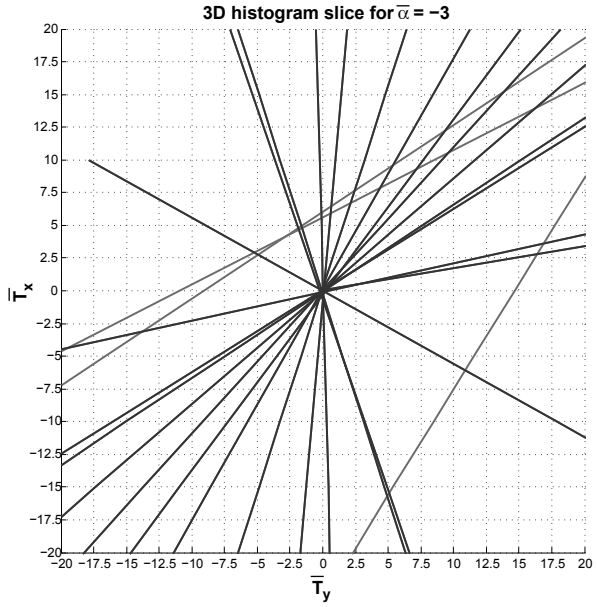


Figure 5.3: A slice of the 3D histogram representing the parameter space (α, T_x, T_y). Each pair of coordinates $(x_{training}, y_{training})$ and (x_{test}, y_{test}) votes for a line in the quantized 2D parameter space $(\overline{T_x}, \overline{T_y})$. Lines corresponding to inliers (blue) intersect in the bin $(\overline{T_x}, \overline{T_y}) = (0,0)$, whereas the remaining lines (red) are related to outliers.

parameters is performed by using the Least Squares algorithm. Outliers have been discarded in the previous steps and, considering only inliers, a good accuracy is achieved.

After recovering the transformation parameters, training and test shapes are aligned and then represented with the Blurred Shape Model [63]. Finally, K-nearest neighbor algorithm with χ^2 measure is used for classification and retrieval purposes (Fig 5.2).

5.1.3.2 Experimental Settings and Results

To properly test the effectiveness of the proposed solution several tests and comparisons with respect to state-of-the-art methods have been performed. In all the tests we considered two different datasets: the public available MPEG-7 CE Shape-1 Part-B dataset which includes 1400 shape samples of 70 different classes, and the 17-class dataset of greylevel symbols used in [63] for testing CSBM. Examples of the shapes contained in both datasets are shown in Fig. 5.4. The original code of SC provided by the authors has been used in our tests. CBSM approach has been reimplemented as described in [63]. The simple combination in cascade of SC and CBSM has been also considered in our comparisons. To limit the problems described in Section 5.1.3.1 (see Fig: 5.1) we have modified the standard parameter setting proposed in [30] in order to limit the variability of the thin plate spline transformation.

Test on each dataset have been repeated five times (5-fold cross-validation procedure) considering 20% samples of each class as test set and the remaining as training set. The 1-NN has

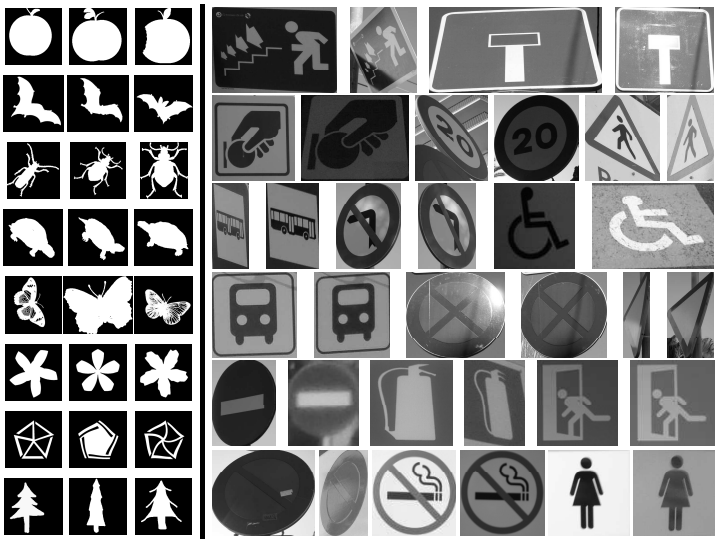


Figure 5.4: Left: examples of eight over seventy different classes of the MPEG-7 dataset. Right: examples of the seventeen different classes of the Symbol dataset.

Table 5.1: Per-Class percentage of correct classified shapes obtained on the 70-class MPEG-7 dataset. The average accuracy of CBSM, SC, the combination of CBSM and SC and the proposed approach are respectively 67.43%, 63.79%, 59.36%, and 76.29%.

METHOD	**apple**	**bat**	**beetle**	**bell**	**bird**	**bone**	**bottle**	**brick**	**butterfly**	**camel**
CBSM [63]	60.00	75.00	50.00	70.00	60.00	75.00	60.00	80.00	70.00	70.00
SC [30]	60.00	60.00	60.00	75.00	85.00	35.00	70.00	75.00	70.00	45.00
SC [30] and CBSM [63]	50.00	60.00	45.00	65.00	75.00	55.00	55.00	70.00	60.00	55.00
Proposed Approach	75.00	65.00	80.00	75.00	65.00	85.00	90.00	85.00	50.00	70.00
METHOD	**car**	**carriage**	**cattle**	**cell phone**	**chicken**	**children**	**chopper**	**classic**	**comma**	**crown**
CBSM [63]	85.00	80.00	80.00	75.00	70.00	70.00	35.00	65.00	75.00	70.00
SC [30]	70.00	70.00	50.00	65.00	80.00	55.00	40.00	75.00	70.00	80.00
SC [30] and CBSM [63]	70.00	70.00	60.00	65.00	75.00	65.00	30.00	70.00	65.00	65.00
Proposed Approach	75.00	60.00	85.00	65.00	80.00	60.00	80.00	60.00	70.00	75.00
METHOD	**cup**	**deer**	**device0**	**device1**	**device2**	**device3**	**device4**	**device5**	**device6**	**device7**
CBSM [63]	75.00	75.00	60.00	30.00	75.00	55.00	45.00	55.00	70.00	75.00
SC [30]	60.00	65.00	45.00	65.00	85.00	65.00	80.00	65.00	90.00	60.00
SC [30] and CBSM [63]	60.00	65.00	45.00	45.00	75.00	50.00	55.00	55.00	75.00	65.00
Proposed Approach	80.00	85.00	70.00	80.00	80.00	85.00	65.00	100.00	80.00	85.00
METHOD	**device8**	**device9**	**dog**	**elephant**	**face**	**fish**	**flatfish**	**fly**	**fork**	**fountain**
CBSM [63]	85.00	85.00	75.00	85.00	60.00	45.00	60.00	85.00	75.00	65.00
SC [30]	50.00	55.00	60.00	55.00	75.00	55.00	65.00	70.00	70.00	70.00
SC [30] and CBSM [63]	55.00	60.00	60.00	65.00	60.00	40.00	60.00	75.00	65.00	60.00
Proposed Approach	65.00	65.00	80.00	85.00	80.00	95.00	80.00	80.00	70.00	70.00
METHOD	**frog**	**glas**	**guitar**	**hammer**	**hat**	**hcircle**	**heart**	**horse**	**horseshoe**	**jar**
CBSM [63]	65.00	60.00	60.00	55.00	70.00	60.00	70.00	65.00	55.00	85.00
SC [30]	55.00	40.00	65.00	55.00	65.00	75.00	55.00	35.00	45.00	50.00
SC [30] and CBSM [63]	55.00	45.00	55.00	50.00	60.00	60.00	60.00	35.00	35.00	60.00
Proposed Approach	80.00	75.00	85.00	70.00	70.00	70.00	75.00	90.00	85.00	70.00
METHOD	**key**	**lizzard**	**lmfish**	**misk**	**octopus**	**pencil**	**personal car**	**pocket**	**rat**	**ray**
CBSM [63]	80.00	85.00	60.00	55.00	55.00	85.00	75.00	70.00	60.00	75.00
SC [30]	70.00	75.00	50.00	75.00	70.00	55.00	50.00	60.00	70.00	90.00
SC [30] and CBSM [63]	65.00	75.00	45.00	55.00	55.00	60.00	55.00	65.00	60.00	75.00
Proposed Approach	70.00	85.00	80.00	60.00	70.00	85.00	75.00	75.00	85.00	70.00
METHOD	**sea snake**	**shoe**	**spoon**	**spring**	**stef**	**teddy**	**tree**	**truck**	**turtle**	**watch**
CBSM [63]	75.00	75.00	65.00	50.00	80.00	50.00	60.00	70.00	70.00	70.00
SC [30]	70.00	80.00	60.00	55.00	80.00	65.00	70.00	70.00	60.00	55.00
SC [30] and CBSM [63]	70.00	70.00	55.00	45.00	80.00	50.00	60.00	70.00	60.00	55.00
Proposed Approach	80.00	70.00	80.00	80.00	85.00	65.00	75.00	70.00	85.00	95.00

been used for classification purposes. Confusion matrices were recorded at each run and the final classification results are obtained averaging on the results of all five runs. The percentage of the

Table 5.2: Per-Class percentage of correct classified shapes obtained on the 17-class Symbol dataset. The average accuracy of CBSM, SC, the combination of CBSM and SC and the proposed approach are respectively 67.62%, 68.29%, 66.42% and 79.12%.

METHOD	**class 1**	**class 2**	**class 3**	**class 4**	**class 5**	**class 6**	**class 7**	**class 8**	**class 9**
CBSM [63]	51.00	56.00	68.00	65.00	60.00	76.00	79.00	65.33	76.00
SC [30]	56.00	56.00	64.00	74.00	72.00	70.00	59.00	77.33	64.00
SC [30] and CBSM [63]	63.00	64.00	60.00	65.00	60.00	71.00	63.00	80.67	68.00
Proposed Approach	87.00	84.00	84.00	87.00	80.00	83.00	75.00	65.33	80.00
METHOD	**class 10**	**class 11**	**class 12**	**class 13**	**class 14**	**class 15**	**class 16**	**class 17**	**Average**
CBSM [63]	80.00	28.00	16.22	42.27	90.00	100.00	96.67	100.00	67.62
SC [30]	80.00	64.00	18.44	51.17	90.00	100.00	96.67	98.33	68.29
SC [30] and CBSM [63]	80.00	52.00	14.00	50.33	75.00	100.00	96.67	100.00	66.42
Proposed Approach	100.00	68.00	26.67	54.27	95.00	100.00	96.67	100.00	79.12

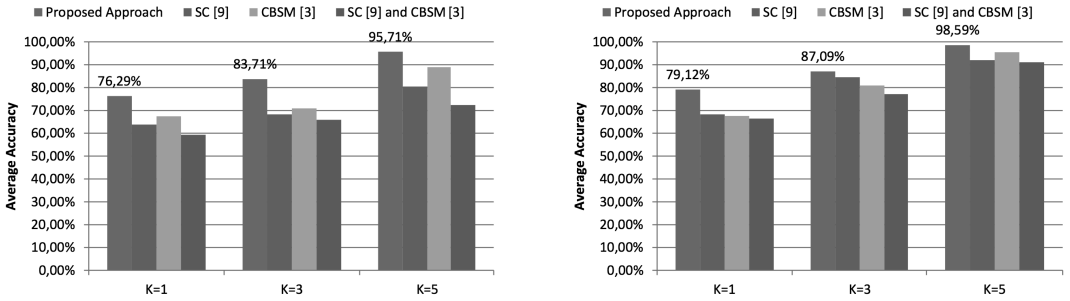
per-class correct classified shapes (diagonal of the final confusion matrix) obtained testing the different approaches on the MPEG-7 dataset is reported in Tab. 5.1, whereas the results obtained on the Symbol dataset are shown in Tab. 5.2. In both cases the proposed strategy outperforms the other state-of-the-art methods obtaining a good margin with respect to the average accuracy. It is worth noting that, considering the MPEG-7 dataset, sometimes the proposed method is not the best one. This behaviour can be explained analysing the intraclass shape deformations that, sometimes, the considered affine model employed in our technique is not able to properly take into account. The 17-class Symbol dataset, considering images of plane objects taken from different points of view can be better modeled by the affine transformation and the proposed approach obtains the best performance in almost all cases (15 over 17). This demonstrates the effectiveness of the proposed framework which exploits the peculiarities of both approaches from which it has origin.

Further tests have been performed to analyze the behaviour of the proposed solution at varying of the parameters of the classifier. Specifically, the performances related to different values of the k parameter of the k-NN classifier have been tested. As expected (see Fig. 5.5) all the considered approaches improve their performances at increasing of the number of neighbours used for classification. The proposed solution outperforms the others in all tests obtaining an average accuracy of 95.71% and 98.59% for MPEG-7 and Symbol datasets respectively when $k = 5$ vs. less than 90% of CBSM and 90.35% of [173].

To properly study the robustness of the considered methods at varying of transformation parameters, several tests have been carried out. An image has been selected from each class and several transformation have been artificially performed (scale, rotation, shear). The average over all classes has been hence reported. The shearing transformation is defined as follows:

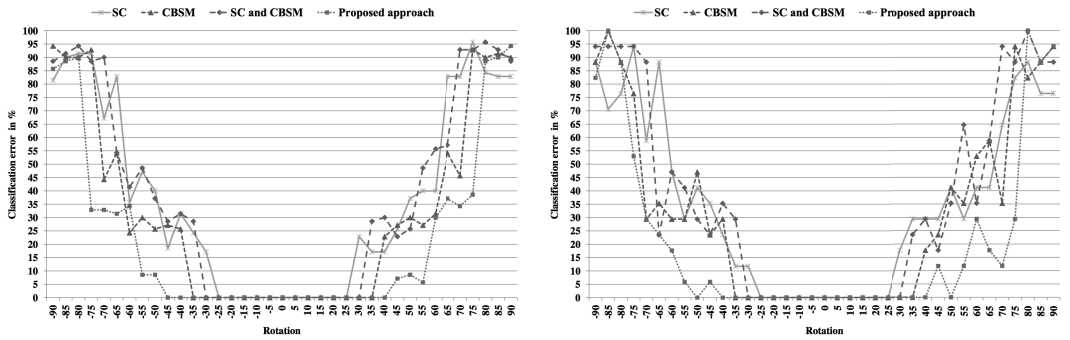
$$x_r = x_s + k y_s \tag{5.5}$$

$$y_r = y_s \tag{5.6}$$



(a) (b)

Figure 5.5: Performance evaluations at varying of the k parameter of k-NN classifier for MPEG-7 (a) and Symbol (b) datasets.



(a) (b)

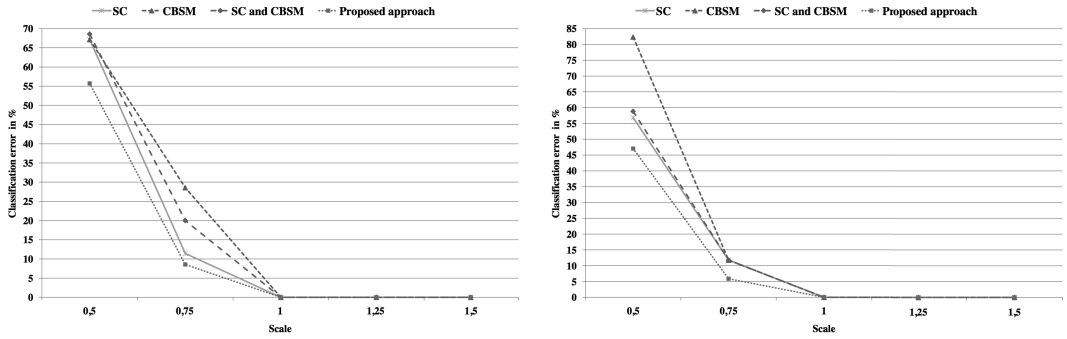
Figure 5.6: Performance evaluation with respect to rotational transformation. The proposed approach outperforms the other methods both on the MPEG-7 CE Shape-1 Part-B dataset (a) and 17-class dataset of greylevel symbols (b).

where (x_s, y_s) and (x_r, y_r) are points in the source image I_s and transformed image I_r respectively, and k is the shear parameter.

As shown in Figs. 5.6, 5.7 and 5.8 the proposed solution obtains satisfactory performances at varying of rotation, scale and shear factor and outperforms the other techniques in all the considered transformations.

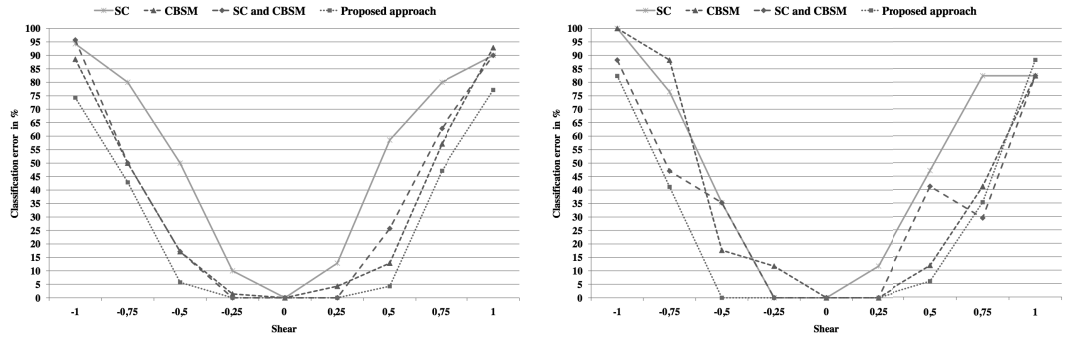
To further demonstrate the effectiveness of the proposed framework, retrieval tests have been accomplished. Considering a specific dataset, for each class an image has been selected as training and the remaining ones as test images to be used to perform the query. This is repeated such that each image belonging to a class is used once in the training set. For example, considering MPEG-7 CE Shape-1 Part-B dataset, we selected 20 pairs of training and test sets made up of 70 and 1330 images respectively.

Each query image has been then associated to a list of training images taking into account the different methods to be compared. The retrieval performance has been evaluated with the



(a) (b)

Figure 5.7: Performance evaluation with respect to scale transformation. The proposed approach outperforms the other methods both on the MPEG-7 CE Shape-1 Part-B dataset (a) and 17-class dataset of greylevel symbols (b).



(a) (b)

Figure 5.8: Performance evaluation with respect to shear transformation. The proposed approach outperforms the other methods both on the MPEG-7 CE Shape-1 Part-B dataset (a) and 17-class dataset of greylevel symbols (b).

probability of the successful retrieval $P(n)$ in a number of test queries:

$$P(n) = \frac{Q_n}{Q} \tag{5.7}$$

where Q_n is the number of successful queries according to $top-n$ criterion, i.e., the correct NDI is among the first n retrieved images, and Q is the total number of queries.

The average of $P(n)$ values with respect to the training and test set pairs is reported in Fig. 5.9. Results show that the proposed solution achieves a good margin of performances with respect to the other techniques on both the MPEG-7 CE Shape-1 Part-B dataset and the 17-class dataset of greylevel symbols. We also show the precision/recall values at $top-n = 1$ in Tab. 5.3. Note that

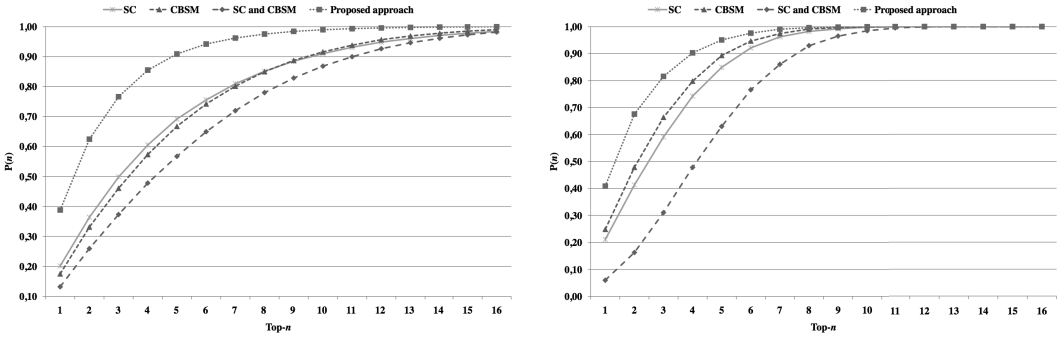


Figure 5.9: Retrieval performance evaluation. The proposed approach achieves the best performances both on the MPEG-7 CE Shape-1 Part-B dataset (a) and the 17-class dataset of greylevel symbols (b).

Table 5.3: Precision/Recall values on the considered datasets.

METHOD	**Precision/Recall MPEG**	**Precision/Recall Symbols**
SC [30]	0.2017	0.2110
CBSM [63]	0.1752	0.2486
SC [30] and CBSM [63]	0.1322	0.0601
Proposed Approach	0.3882	0.4088

the precision and recall for $top - n = 1$ are equivalent because there is only one correct match for each query.

5.1.3.3 Classification Performances on a Large Sketches Dataset

The experiments performed so far demonstrated the performances and the robustness of the proposed method with respect to geometric variability on two classic datasets. To further assess the proposed technique we have performed further tests on a large hand sketches dataset composed by 20,000 examples distributed over 250 object categories [62]. Some examples of sketch sampled at random from the dataset are shown in Fig. 5.10. At the best of our knowledge, this is the larger labeled dataset of shapes on which the proposed method can be tested. The dataset covers most objects of everyday life. Each category is recognizable from its shape alone and does not require a context for recognition and each class is specific enough for testing purpose (i.e., it does not contain general classes composed by subcategories such as “Animal” or “musical instrument”). Humans are able to recognize correctly on average 73.1% percent of all 20,000 sketches.

In [62] a method based on bag of feature paradigm (BoW) has been presented. Local histograms of orientations have been considered as main local feature. Both variants hard BoW and soft BoW have been tested considering k-NN and SVM as classifiers. To test their approach the authors used a 3-fold cross-validation method. The final classification accuracy have been

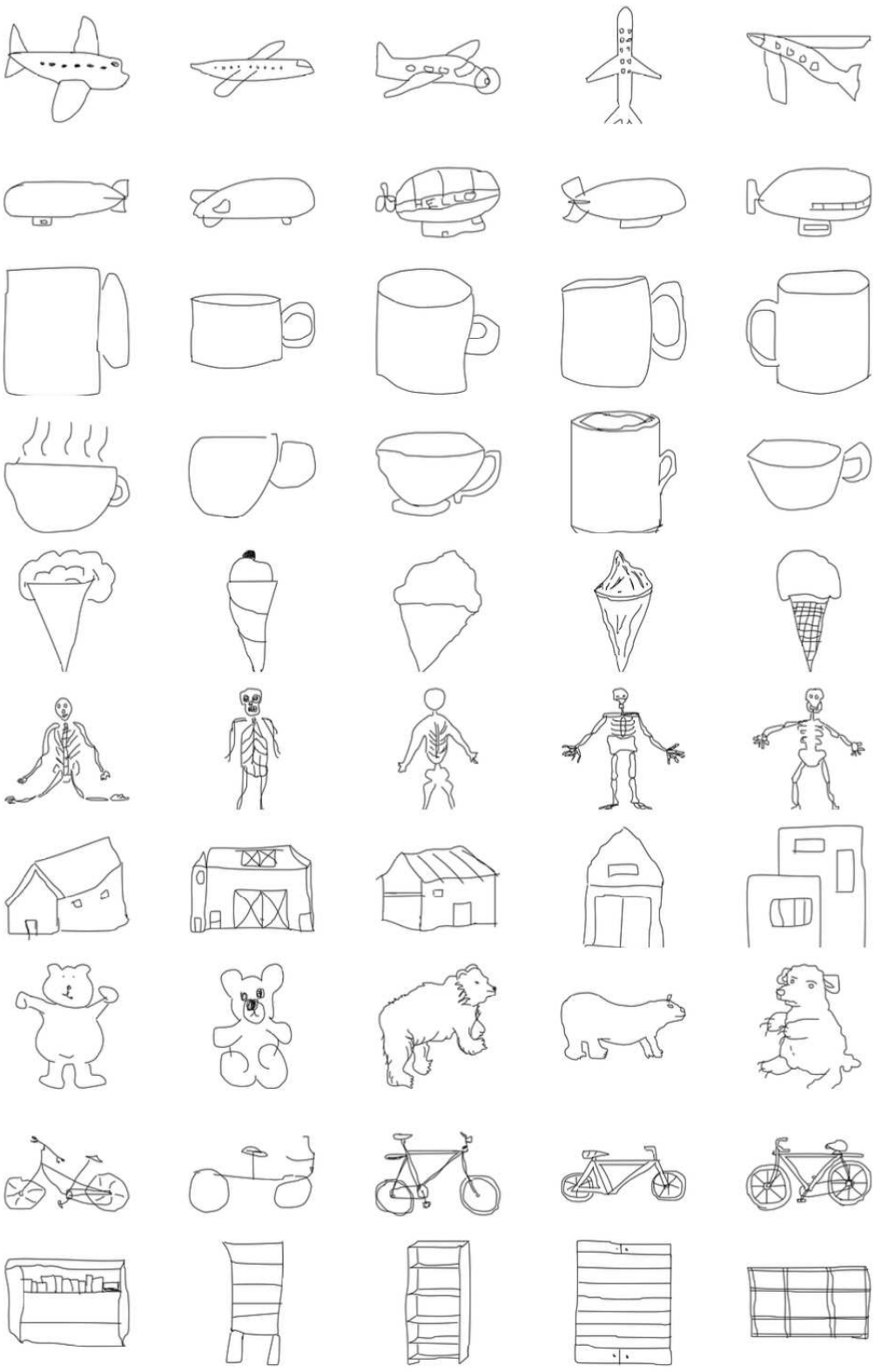


Figure 5.10: Examples of shapes of the hand sketch dataset [62]. Rows correspond to the following ten categories: *airplane*, *blimp*, *mug*, *cup*, *ice-cream-cone*, *human-skeleton*, *barn*, *bear*, *bicycle*, *bookshelf*. The dataset presents high within variability (see 8^{th} row) as well as contains classes with low between vaiability (see 2^{nd} and 3^{rd} rows).

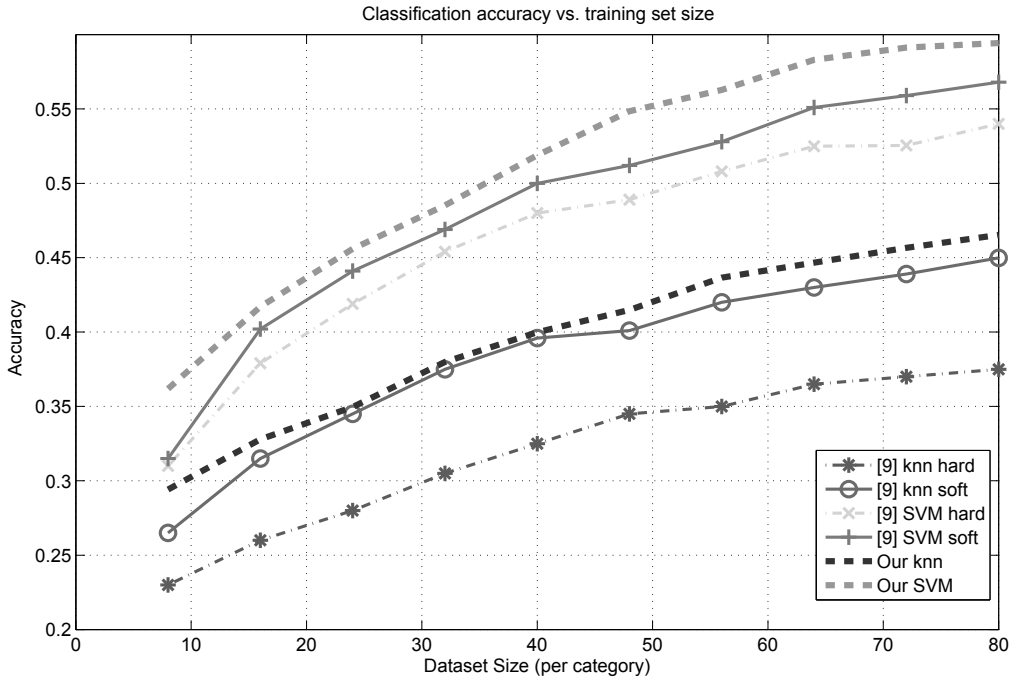


Figure 5.11: Classification accuracy comparison between the proposed approach (Our) Vs. [62] in both k-NN and SVM on growing data-set size.

obtained averaging over the 3 runs. We compared the proposed approach with respect to the one proposed in [62]. We used a k-NN classifier with k=4, and SVM in one against all modality to run our tests after representing sketches with our method. As shown in 5.11 the proposed approach outperforms [62] in both k-NN and SVM versions.

5.1.4 A 3D approach to ballistics

In previous Section a robust shape alignment technique for 2D shapes has been described. The technique exploits a dictionary of features in order to understand which transformation to apply to two shapes for the alignment and recognition task. A robust alignment and classification technique is indeed needed for the 3D cartridge comparison task too. As stated at the beginning of this chapter the analysis is done by the forensics examiner with the aim of selecting the most relevant and useful geometrics for the task of comparison.

The "Reparto Investigazioni Scientifiche di Messina" (RIS) is a LEA specialized in scientific investigation. They provided a dataset of 3D point clouds from fired cartridges. The acquisition was carried out with an high-level laser scanner with a precision of 0.5 micron. The dataset consists of 2 point clouds per 35 firearms, for a total of 70 point clouds. Each obtained point cloud is dense and is a digital representation of a cartridge in STL point cloud format. In this representation all the geometrical information useful for cartridge comparison are present. But

discovering the most important or "salient" for the task of comparison is trivial. Thus just applying 3D point cloud alignment techniques like ICP [31, 48] does not achieve results that are worth discussing.

Results can be improved by selecting a subset of points: the most informative points about the geometrical imprintings left by the firearm. This manual approach in conjunction with ICP alignment algorithm improves a bit the results but a very deep knowledge of the domain is needed for selecting the right points in order to achieve best results. Moreover sometimes these points or set of points are not visible in the point cloud. In fact the manual selection does not take into account that some relevant geometry properties could be irrelevant for the task of alignment and classification due to the thousands of variables, physical an kinetic, that come into play when a bullet is fired.

The proposed technique exploits a Convolutional Neural Network (CNN) with the aim of finding the best filters on the input point clouds in order to select only the most relevant points (keypoints) for the alignment and classification task. While CNN are now used in particular for the object detection and recognition tasks, they have been invented to learn the best filtering on images for encode or filter information.

At first a pre-processing is needed to identify the ‚Äúvalid areas‚Äù and eliminating the ‚Äúinvalid areas‚Äù from consideration. In Figure 5.12 valid area vs. invalid one is shown.

5.1.4.1 Preprocessing

Given a Point Cloud (PC) representing a Cartridge surface, in which a negative impression from a firearm is visible and a reference system is defined as in Figure 5.12, we look for a centroid C defined as:

$$C = \frac{\sum_{i=0}^{n} X_i}{n}, \tag{5.8}$$

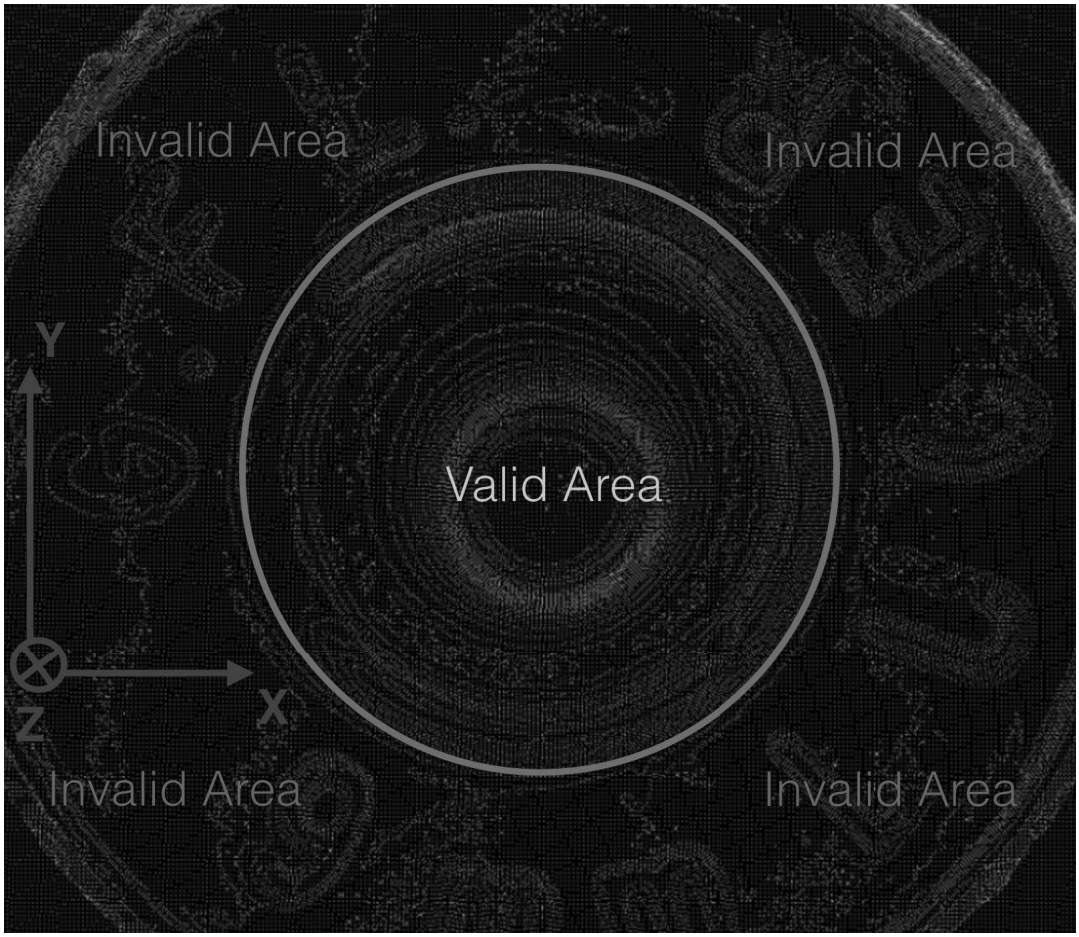
Where n is the number of points in the PC and X_i are i-th point coordinates. We than look for the projection along the Z-Axis from C to the PC which will identify another point C_o on the PC. The C_o is almost central in the lowest part of the PC with respect to the Z-Axis.

Given C_o and defined a radius r arbitrary little, it is possible to find the neighbors of C_o as P. Given a $\alpha \in \{0, 90, 180, 270\}$ a direction defined in the XZ-plan, it is possible to identify the nearest point in P to C_o along that direction. Thus iterating this operation for the found point P_α and its neighbors, for each direction a walk W_α on the surface of the point cloud from C_o can be defined.

Given W_α a set of point describing the walk from C_o along a direction on the PC surface, for each point $w \in W_\alpha$ is possible to find a normal to the plane identified by the neighbors of w.

We define NP_α the point in the walk W_α for which the normal computed on the plane defined by its neighbors satisfies the following condition of being parallel to the normal of XZ-plan plus an arbitrary ϵ.

Figure 5.12: A point cloud representing the examined cartridge surface. A reference system is defined in the point cloud. In figure are shown areas identifiable on cartridges. The valid area (green) brings information for the ballistics task. The invalid one (red) presents noise and could reduce the accuracy of overall matching techniques.



A radius R will than be defined as follows:

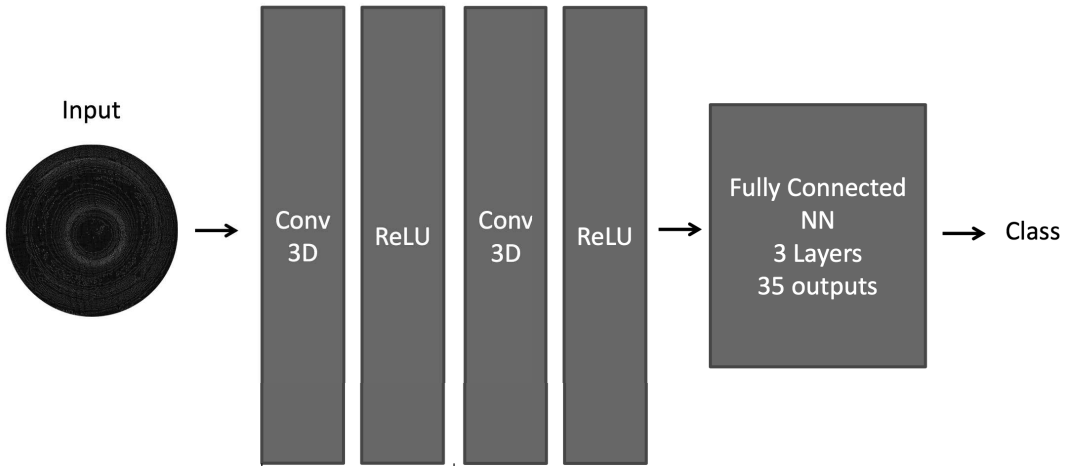
$$R = \min_{\forall \alpha} |C_o - NP_\alpha| \tag{5.9}$$

Now a sphere S can be defined in the PC with center C_o and radius R. All point of PC that are inside the sphere are in the valid area. The others are in the invalid area. The valid area constitutes a new point cloud PC_d that can be used as input to the CNN in order to find the most relevant set of point for the alignment and classification task.

5.1.4.2 Alignment and classification of 3D point clouds

The task of the entire approach is to find the most relevant points in order to to the best alignment and the most accurate classification of the firearm (device and model) who fired the analysed

Figure 5.13: Convolutional Neural Network architecture for the firearm classification task from cartridges point clouds.



cartridge. In order to do this cartridges from different firearms are to be compared. Once a valid area on the Point Cloud is identified a Convolutional Neural Network on the dataset can be trained. The designed CNN has the architecture described in figure 5.13.

The output of the last layer has a soft-max classifier that assigns a class between the considered 35 to the input PC. The designed CNN tries to learn best weights for each layer in order to achieve best classification results. Certainly the CNN finds the best solution on the very little dataset provided with over-fitting problems. For this reason, the results of this classification technique is not worth discussing. The most interesting information learnt by the CNN is indeed those weights that is some way represent a level of relevance for each point in the PC for the task of firearm classification. Once the CNN is trained we than use only the first layer in order to produce a new PC with only those points relevant for the classification task. This is not the first time a CNN is used with this intent [184, 185].

Once the input PC is processed by the first layer of the CNN a reduced set of points is identified and can now be used with an alignment algorithm like ICP. Results are reported in table 5.4.

ICP is not the best solution for alignment, it is possible to apply the technique described in Section 5.1.3 on the overall PC in order to build the dictionary then it is possible to describe the CNN-discovered keypoints and the apply the shape alignment technique to all the PC. Results of the combined technique are shown in table 5.4.

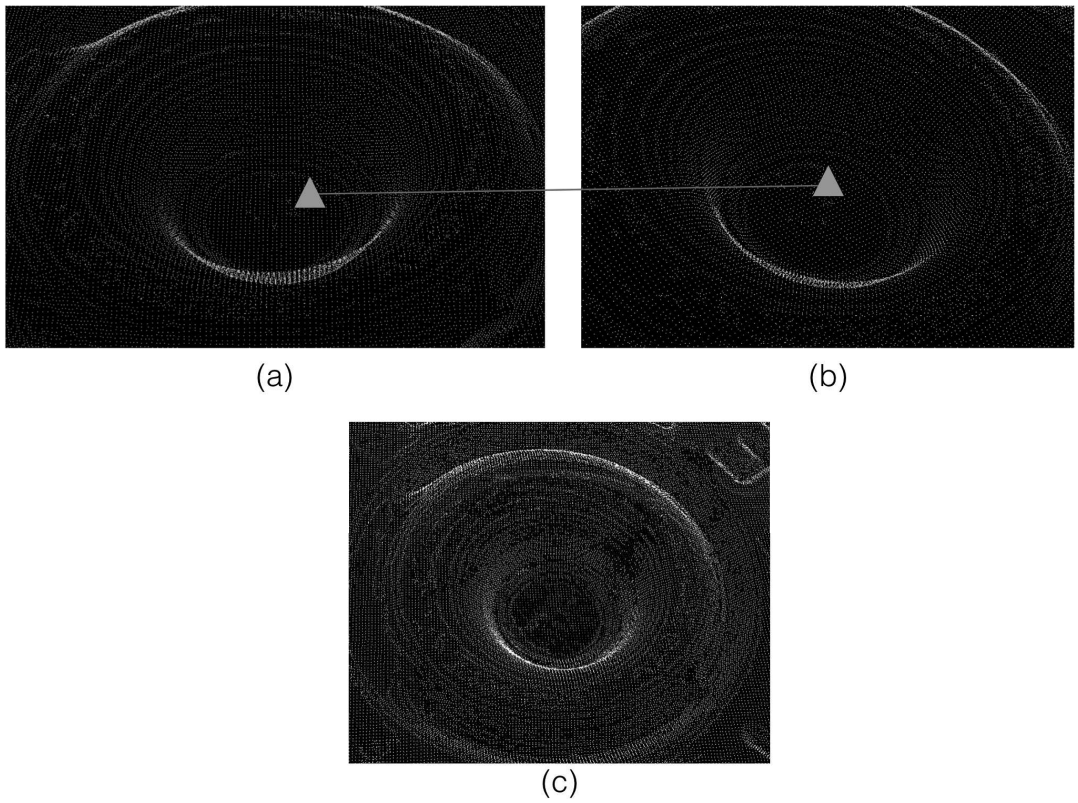
The technique described in 5.1.3 was also applied without the CNN-based point selection and the results are shown in table 5.4.

Figure 5.14 shows the results of the alignment.

Table 5.4: Firearm classification results with different alignment and metric techniques.

Technique	Accuracy
ICP Simple	51%
CNN Keypoint Extraction + ICP	86%
Shape Alignment Simple	71%
CNN Keypoint Selection + Shape Alignment	92%

Figure 5.14: The Point cloud matching results. (a) and (b) are two point clouds representing a cartridge fired from the same firearm. The orange triangle is one of the identified keypoints that is found as a match between the two point clouds. (c) shows the alignment result of the two point clouds.



5.1.5 Firearm Serial Number Reconstruction

The problem of automatic numeric digit recognition from a digital image has been widely addressed and dealt with within the broader field of Computer Vision research, as it represents the ability to automate a process that the human brain can perform very efficiently and in very short time. Of greater importance is the ability to recognize a number when it is not fully visible because it is affected by noise or presents missing parts. This is a process that the human brain can also perform quite well, while for an automatic system the problem is more complex. It is

therefore necessary to have a system that not only can precisely recognize a number represented in different ways and with different styles, but it also suffers from the lack of information due to noise. The importance of numerical recognition is due to the fact that they can represent, for example, a unique serial number that for some reason, is not immediately recognizable but whose identification is very important In some application domain. In particular, this is the case with law enforcement officers who often find themselves working with firearms found in the crime scene, which have imperfections on the unique serial number. These imperfections, perhaps created in order to prevent them from being identified, can be of a variety of nature and can lead to make serial number totally unreadable. The serial number can be tampered with: abrasion, the use of a drill to pierce the affected area or the use of chemicals.

There are techniques that physically allow to exalt the numeric digits of the serial number of the weapon, for example through the application of particular chemical agents [55], but an automatic software system that, starting from an High-Resolution image of the serial number, returns a set of possible values with the corresponding probability turns out to be very useful as forensic evidence or for investigative purposes. In fact, even if the serial number in question is recognizable with a certain confidence by a human operator, in such a subjective estimate court it might not be enough to prove how wanted. Moreover the image analysis process does not alter the evidence and is repeatable.

The purpose of this proof of concept is demonstrating that a software can be capable of recognizing and identifying digits of a serial number, within a certain range of probabilities, starting from the images representing firearms. The goal of the application is to get the probability that the serial number displayed is a certain serial rather than another. To realize this software, machine learning techniques where exploited in order to traine on a specific database and be able to automate the identification of digits of a serial number.

5.1.5.1 The proposed technique

Starting from a dataset of only 11 elements, provided by The "Reparto Investigazioni Scientifiche di Messina" (RIS) and presenting different types of erasure of the serial number. And given another set of 120 images representing intact serial numbers, this tiny dataset make impossible to define a machine learning technique able to solve the problem, since the training dataset can not be sufficiently representative of the universe. Therefore, the proposed technique represents a Proof of Concept that describes a starting point for a new track for which, in state of the art to date, there are still no results. In order to build a serial number reconstruction engine, starting from the few available data on images representing physically erased information on a firearm we must first train a classifier in recognizing the digits. To do this, a deep neural network is trained on the MNIST dataset: a database containing a training set of 60000 examples of black and white images representing free handwriting digit numbers from 0 to 9 and 10000 examples to test algorithms. It is a subset of the largest NIST database in which the elements have been

standardized, centered and scaled to a standard size of 28x28 pixels. The database in question is very much used because it allows you to get more accurate estimates thanks to the large intra-class variance and low inter-class variance of its elements.

A Convolutional Neural Network was designed to solve the task of MNIST digit recognition. The model of the CNN is described in [**?**] and achieves an accuracy on test-set of about 97%. Given the trained CNN on MNIST dataset, a new convolutional layer is defined in order to learn how to enhance and transform the input image into one on which the MNIST-trained CNN can recognize digits even if partially or totally erased.

The new Convolutaion Layer receives in input each digit of the input image, enhance the input digit in order and outputs the result to the MNIST-trained CNN that predicts the corresponding digits. The new Convolutional Layer needs update its weights and to do this is defined as follows.

Given a Training Set S of images with dimensions 28x28.

At each training step each element of the training set S is convolved with W

$$R = (S * W_c + b), \tag{5.10}$$

Where W_c is a Tensor of weights updated after digits classification with dimensions (n,28,28,32) where n is the number of training data, b is the bias factor.

From R it is possible to reconstruct a new image as follows:

$$G_i(x,y) = \frac{\sum_{j=1}^{32} R_i(x,y)_j}{32} \forall i \in 1,...,n \forall x,y \in 1,...,28 \tag{5.11}$$

Values are then normalized as follows:

$$N_i(x,y) = 255 * \frac{G_i(x,y) - \min G_i}{\max G_i - \min G_i} \forall i \in 1,...,n \forall x,y \in 1,...,28 \tag{5.12}$$

Thus binary images are obtained by applying the Otsu algorithm to N_i images and then used as input on the MNIST-trained CNN. If the classification is wrong, W_c weights are then updated as follows:

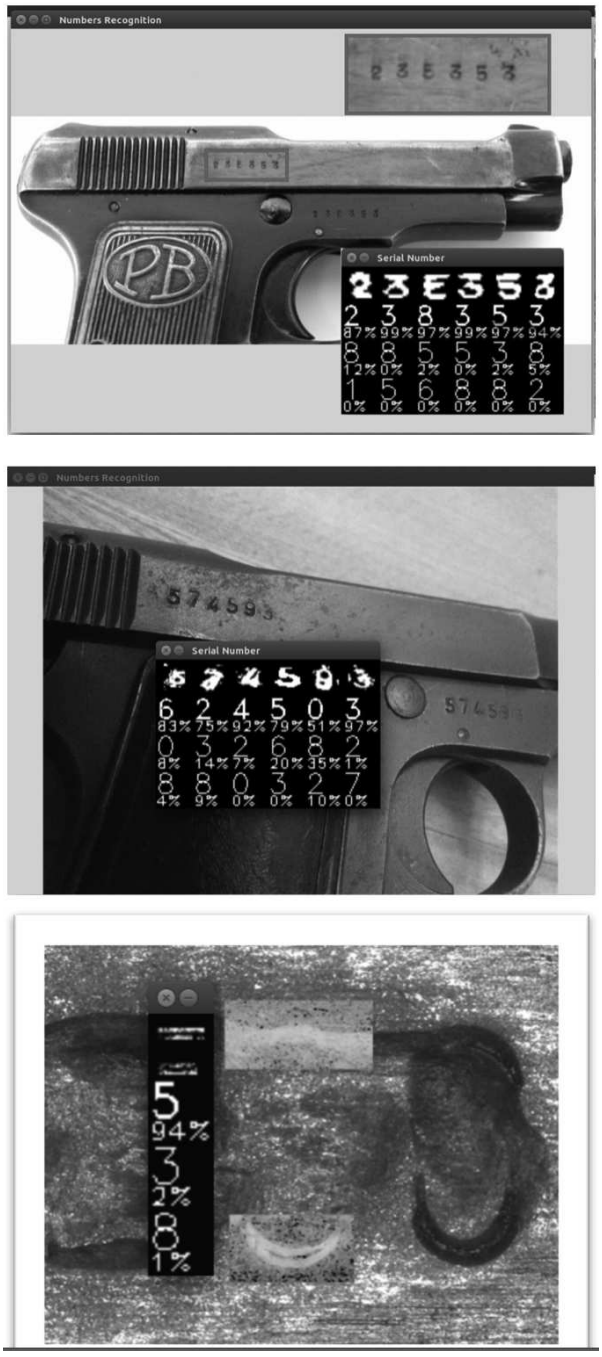
$$W_c = \begin{cases} \frac{255}{B_i(x,y)} B_i(x,y) > 128 \\ \frac{B_i(x,y)}{255} B_i(x,y) < 128 \end{cases} \tag{5.13}$$

The results obtained after 1000 training iterations are shown in Figure 5.15. The preliminary results are promising but a more accurate and detailed procedure need to be developed on a much bigger dataset. Moreover a specific technique should be developed for each kind of Serial Number alteration.

The Serial Number reconstruction is just an application of other useful automatic forensics task like Driving License Plate recognition and reconstruction and other many tasks for which the original information is reduced, concealed or with low quality and a reconstruction technique is needed to extract new information useful for investigation purposes.

Future works are needed on the task by exploiting Neural Network and Autoencoders techniques.

Figure 5.15: Serial Number Recognition results on firearms.



CHAPTER 6

CONCLUSIONS

In this work the Ballistics problem has been addressed. Defined Ballistics as the reconstruction of the history of an evidence starting from the traces of the device that fired (acquired) the evidence itself it is possible to extend the term both to device source identification for images and Classic Ballistics examination for firearms. This two parts of the meaning of Ballistics have been analysed with novel techniques addressing the task in complex scenarios like Social Networks for the image domain. A new multimedia-based technique has been proposed for both ballistics from cartridges and firearms canceled serial number reconstruction. Results are promising but, for forensics application, a much more general specification have to be defined to be shared and accepted as a common analysis technique by the forensics community. To do this much bigger dataset are needed to prove and benchmark the techniques proposed in this work against the techniques that will be proposed by other scientists of the field. Only by having a shared database for each problem and a portfolio of techniques benchmarked on it there will be applications of Digital Forensics acceptable in court law. This is a common issue for each field of Digital Forensics.

BIBLIOGRAPHY

[1] *BOWS 2*.
http://bows2.ec-lille.fr/.

[2] *CASIA*.
http://forensics.idealtest.org/.

[3] *DJPEG – LibJPEG open-source project on GITHUB*.
https://github.com/LuaDist/libjpeg.

[4] *Kodak Gallert*.
http://r0k.us/graphics/kodak/.

[5] *MAVEN*.
http://maven-project.eu/.

[6] *NRCS Gallery*.
http://photogallery.nrcs.usda.gov/res/sites/photogallery/.

[7] *REVEAL*.
http://revealproject.eu/.

[8] *REWIND*.
http://www.rewindproject.eu/.

[9] *S-Five*.
https://www.s-five.eu/mediawiki/index.php.

[10] K. S. A.:, *The discrete cosine transform (DCT): theory and application*, Michigan State University, 2003.

[11] P. A.:, *An overview on image forensics*, ISRN Signal Processing, 2013.

[12] N. AHMED, T. NATARAJAN, AND I. T. O. C. RAO K. R.: DISCRETE COSINE TRANSFORM, *100*, (1974), pp. 90–93.

[13] A. N. AKANSU AND H. P. R. M. SIGNAL DECOMPOSITION: TRANSFORMS, *subbands*, and wavelets, Academic Press, 2000.

[14] N. ALLAN, L. PAN, AND A. XIANG Y.:, *Novel method for detecting double compressed facebook JPEG images*, Applications and Techniques in Information Security, (2014), pp. 191–198.

[15] I. AMERINI, L. BALLAN, R. CALDELLI, A. DEL BIMBO, AND G. SERRA, *Geometric tampering estimation by means of a sift-based forensic analysis*, in Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, IEEE, 2010, pp. 1702–1705.

[16] I. AMERINI, R. CALDELLI, P. CRESCENZI, A. DEL MASTIO, AND A. MARINO, *Blind image clustering based on the normalized cuts criterion for camera identification*, Signal Processing: Image Communication, 29 (2014), pp. 831–843.

[17] G. AZZARO, M. CACCAMO, J. FERGUSON, S. BATTIATO, G. M. FARINELLA, G. GUARNERA, G. PUGLISI, R. PETRIGLIERI, AND G. LICITRA, *Objective estimation of body condition score by modeling cow body shape from digital images*, Journal Dairy Science, 94 (2011), pp. 2126–2137.

[18] W. A. B.:, *Image compression using the Discrete Cosine Transform*, Mathematical Journal, 4 (1994), pp. 81–88.

[19] X. BAI, W. LIU, AND Z. TU, *Integrating contour and skeleton for shape classification*, in IEEE International Conference on Computer Vision Workshops, 2009, pp. 360–367.

[20] M. BARNI, A. COSTANZO, AND S. L.:, *Identification of cut & paste tampering by means of double-JPEG detection and image segmentation*, in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS, 2010, pp. 1687–1690.

[21] M. BARNI AND F. PÉREZ-GONZÁLEZ, *Coping with the enemy: Advances in adversary-aware signal processing*, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 8682–8686.

[22] S. BATTIATO, G. M. FARINELLA, O. GIUDICE, AND G. PUGLISI, *Aligning bags of shape contexts for blurred shape model based symbol classification*, in Proceedings of International Conference on Pattern Recognition (ICPR), 2012, pp. 1598–1601.

[23] S. BATTIATO, G. M. FARINELLA, E. MESSINA, AND G. PUGLISI, *Robust image alignment for tampering detection*, IEEE Transactions on Information Forensics and Security, 7 (2012), pp. 1105–1117.

[24] S. BATTIATO, G. M. FARINELLA, E. MESSINA, AND G. PUGLISI, *Robust image alignment for tampering detection*, IEEE Transactions on Information Forensics and Security, 7 (2012), pp. 1105–1117.

[25] S. BATTIATO AND M. G.:, *Digital forgery estimation into DCT domain: a critical analysis*, in Proceedings of the First ACM workshop on Multimedia in forensics, 6, 2010, 25, pp. 389–399.

[26] S. BATTIATO, O. GIUDICE, AND A. PARATORE, *Multimedia forensics: discovering the history of multimedia contents*, in Proceedings of the 17th International Conference on Computer Systems and Technologies 2016, ACM, 2016, pp. 5–16.

[27] S. BATTIATO, M. MANCUSO, A. BOSCO, AND M. GUARNERA, *Psychovisual and statistical optimization of quantization tables for dct compression engines*, in Image Analysis and Processing, 2001. Proceedings. 11th International Conference on, IEEE, 2001, pp. 602–606.

[28] S. BATTIATO AND G. MESSINA, *Digital forgery estimation into DCT domain: A critical analysis*, in Proceedings of the First ACM Workshop on Multimedia in Forensics, MiFor '09, New York, NY, USA, 2009, ACM, pp. 37–42.

[29] S. BAYRAM, H. SENCAR, N. MEMON, AND I. AVCIBAS, *Source camera identification based on cfa interpolation*, in Image Processing, 2005. ICIP 2005. IEEE International Conference on, vol. 3, IEEE, 2005, pp. III–69.

[30] S. BELONGIE, J. MALIK, AND J. PUZICHA, *Shape matching and object recognition using shape contexts*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (2002), pp. 509–522.

[31] P. J. BESL, N. D. MCKAY, ET AL., *A method for registration of 3-d shapes*, IEEE Transactions on pattern analysis and machine intelligence, 14 (1992), pp. 239–256.

[32] P. BESTAGINI, A. ALLAM, S. MILANI, M. TAGLIASACCHI, AND S. TUBARO, *Video codec identification*, in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 2257–2260.

[33] P. BESTAGINI, S. MILANI, M. TAGLIASACCHI, AND S. TUBARO, *Local tampering detection in video sequences*, in Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on, IEEE, 2013, pp. 488–493.

[34] T. BIANCHI AND P. A.:, *Detection of non-aligned double JPEG compression with estimation of primary compression parameters*, in 18th IEEE International Conference on Image Processing (ICIP, 2011, pp. 1929–1932.

[35] T. BIANCHI AND P. A., *Image forgery localization via block-grained analysis of JPEG artifacts*, IEEE Transactions on Information Forensics and Security, 7 (2012), pp. 1003–1017.

[36] T. BIANCHI AND A. DE ROSA, *and piva a.: Improved DCT coefficient analysis for forgery localization in JPEG images*, in Conference on Acoustics Speech and Signal Processing, I. International, ed., 2444, 2447, 2011, ICASSP).

[37] L. BIN, *and ng t.-t. and li x. and tan s. and huang j.: JPEG noises beyond the first compression cycle*, technical report, TR2014-001, Shenzhen University, widrow1996statistical, 2014.

[38] X. BO, W. JUNWEN, L. GUANGJIE, AND D. YUEWEI, *Image copy-move forgery detection based on surf*, in Multimedia information networking and security (MINES), 2010 international conference on, IEEE, 2010, pp. 889–892.

[39] R. BÖHME AND M. KIRCHNER, *Counter-forensics: Attacking image forensics*, in Digital Image Forensics, Springer, 2013, pp. 327–366.

[40] S. BRAVO-SOLORIO AND A. K. NANDI, *Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics*, Signal Processing, 91 (2011), pp. 1759–1770.

[41] A. R. BRUNA, G. MESSINA, AND S. BATTIATO, *Crop Detection through Blocking Artefacts Analysis*, vol. 6978 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 650–659.

[42] T. CAGLAR, Y. BERRIN, AND S. T. METIN, *Sketched symbol recognition with auto-completion*, Pattern Recognition, 45 (2012), pp. 3926–3937.

[43] R. CALDELLI, R. BECARELLI, AND I. AMERINI, *Image origin classification based on social network provenance*, IEEE Transactions on Information Forensics and Security, 12 (2017), pp. 1299–1308.

[44] G. CAO, Y. ZHAO, AND R. NI, *Edge-based blur metric for tamper detection*, Journal of Information Hiding and Multimedia Signal Processing, 1 (2010), pp. 20–27.

[45] G. CAO, Y. ZHAO, R. NI, L. YU, AND H. TIAN, *Forensic detection of median filtering in digital images*, in Multimedia and Expo (ICME), 2010 IEEE International Conference on, IEEE, 2010, pp. 89–94.

[46] A. CASTIGLIONE, G. CATTANEO, AND A. DE SANTIS, *A forensic analysis of images on online social networks*, in Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on, IEEE, 2011, pp. 679–684.

[47] A. CASTIGLIONE, G. CATTANEO, AND A. D. SANTIS, *A forensic analysis of images on online social networks*, Intelligent Networking and Collaborative Systems, International Conference on, 0 (2011), pp. 679–684.

[48] Y. CHEN AND G. MEDIONI, *Object modelling by registration of multiple range images*, Image and vision computing, 10 (1992), pp. 145–155.

[49] Y. CHEN AND V. L. THING, *A study on the photo response non-uniformity noise pattern based image forensics in real-world applications*, in Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), 2012, p. 1.

[50] P. COMESANA-ALFARO AND P.-G. F.:, *Optimal counterforensics for histogram-based forensics*, in Proc. IEEE Int, Speech, and Signal Process, 2013, Conf. Acoust, pp. 3048–3052.

[51] F. D. O. COSTA, E. SILVA, M. ECKMANN, W. J. SCHEIRER, AND A. ROCHA, *Open set source camera attribution and device linking*, Pattern Recognition Letters, 39 (2014), pp. 92–101.

[52] I. J. COX AND J.-P. M. LINNARTZ, *Public watermarks and resistance to tampering*, in International Conference on Image Processing (ICIP,Äô97), 1997, pp. 26–29.

[53] D. COZZOLINO, D. GRAGNANIELLO, AND L. VERDOLIVA, *Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques*, in Image Processing (ICIP), 2014 IEEE International Conference on, IEEE, 2014, pp. 5302–5306.

[54] L. DA FONTOURA COSTA AND R. M. CESAR JR., *Shape Classification and Analysis: Theory and Practice*, CRC Press, Inc., Boca Raton, FL, USA, 2nd ed., 2009.

[55] L. DA SILVA AND P. A. M. DOS SANTOS, *Recovering obliterated laser engraved serial numbers in firearms*, Forensic Science International, 179 (2008), pp. e63–e66.

[56] M. R. DALIRI AND V. TORRE, *Robust symbolic representation for shape recognition and retrieval*, Pattern Recognition, 41 (2008), pp. 1782–1798.

[57] X. J. S. DATABASE:, *Large-scale scene recognition from abbey to zoo*, in IEEE conference on Computer vision and pattern recognition (CVPR, 2010, pp. 3485–3492.

[58] C. DC-008, *Exchangeable image file format for digital still cameras: Exif version 2.3*, (2012).

[59] L. DEBIASI AND A. UHL, *Blind biometric source sensor recognition using advanced prnu fingerprints*, in Signal Processing Conference (EUSIPCO), 2015 23rd European, IEEE, 2015, pp. 779–783.

[60] F. DEVERNAY AND O. FAUGERAS, *Automatic calibration and removal of distortion from scenes of structured environments*, in Investigative and Trial Image Processing, vol. 2567, SPIE, 1995.

[61] A. E. DIRIK AND A. KARAKÜÇÜK, *Forensic use of photo response non-uniformity of imaging sensors and a counter method*, Optics express, 22 (2014), pp. 470–482.

[62] M. EITZ, J. HAYS, AND M. ALEXA, *How do humans sketch objects?*, ACM Transactions on Graphics (SIGGRAPH), 31 (2012), pp. 44:1–44:10.

[63] S. ESCALERA, A. FORNÉS, O. PUJOL, J. LLADÓS, AND P. RADEVA, *Circular blurred shape model for multiclass symbol recognition*, IEEE Transactions on Systems, Man, and Cybernetics, 41 (2011), pp. 497–506.

[64] H. F.:, *Detecting double JPEG compression with the same quantization matrix*, IEEE Transactions on Information Forensics and Security, 15 (2005), pp. 27–38.

[65] M. F.:, *Distribution shape of two-dimensional DCT coefficients of natural images*, Electronics Letters, 29 (1993), pp. 1935–1936.

[66] W. FAN, K. WANG, F. CAYRE, AND X. Z.:, *JPEG anti-forensics using non-parametric DCT quantization noise estimation and natural image statistics*, in Proceedings of the first ACM workshop on Information hiding and multimedia security, 2013, pp. 117–122.

[67] W. FAN, K. WANG, F. CAYRE, AND X. Z., *JPEG anti-forensics with improved trade-off between forensic undetectability and image quality*, IEEE Transactions on Information Forensics and Security, 9 (2014), pp. 1211–1226.

[68] Z. FAN AND DE QUEIROZ R. L.:, *Introduction to fourier analysis and wavelets*, IET Computer Vision, 5 (2011), pp. 320–334.

[69] Z. FAN AND DE QUEIROZ RICARDO L.:, *Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history*, in Proceedings of the International Conference on Image Processing, 1, 2000, pp. 948–951.

[70] Z. FAN AND DE QUEIROZ RICARDO L., *Identification of bitmap compression history: JPEG detection and quantizer estimation*, IEEE Transactions on Image Processing, 12 (2003), pp. 230–235.

[71] H. FARID, *Digital image ballistics from JPEG quantization: A followup study*, Department of Computer Science, Dartmouth College, Tech. Rep. TR2008-638, (2008).

[72] G. M. FARINELLA, G. IMPOCO, G. GALLO, S. SPOTO, G. CATANUTO, AND M. B. NAVA, *Objective outcome evaluation of breast surgery*, in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006, vol. 4190 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 776–783.

[73] G. M. FARINELLA AND B. S.:, *Scene classification in compressed and constrained domain, computer vision*, IET, 5 (2011), pp. 320–334.

[74] L. Z. FAST, *automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis*, Pattern Recognition, 42 (2009), pp. 2492–2501.

[75] X. FENG AND I. E. I. DOËRR G.: JPEG RECOMPRESSION DETECTION, *75410j*, (2010).

[76] A. FISCHER H.:, *history of the central limit theorem: from classical to modern probability theory*, Springer, Science & Business Media, 2010.

[77] M. FONTANI, E. ARAGONES-RUA, C. TRONCOSO, AND B. M.:, *The watchful forensic analyst: Multi-clue information fusion with background knowledge*, IEEE International Workshop on Information Forensics and Security (WIFS), (2013), pp. 120–125.

[78] M. FONTANI, T. BIANCHI, A. DE ROSA, A. PIVA, AND M. BARNI, *A framework for decision fusion in image forensics based on dempster–shafer theory of evidence*, IEEE Transactions on Information Forensics and Security, 8 (2013), pp. 593–607.

[79] M. FONTANI, A. BONCHI, AND P. A.:, *Countering anti-forensics by means of data fusion*, IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 9028 (2014).

[80] D. FU, Y. Q. SHI, AND A. SU W.:, *generalized benford's law for JPEG coefficients and its applications in image forensics*, Electronic Imaging, 6505 (2007).

[81] S. G.:, *The discrete cosine transform*, SIAM review, 41 (1999), pp. 135–147.

[82] F. GALVAN, G. PUGLISI, A. R. BRUNA, AND S. BATTIATO, *First quantization matrix estimation from double compressed JPEG images*, IEEE Transactions on Information Forensics and Security, 9 (2014), pp. 1299–1310.

[83] F. GALVAN, G. PUGLISI, A. R. BRUNA, AND B. S.:, *First quantization coefficient extraction from double compressed JPEG images*, in International Conference on Image Analysis and Processing (ICIAP, 2013, pp. 783–792.

[84] F. GALVAN, G. PUGLISI, A. R. BRUNA, AND B. S., *First quantization matrix estimation from double compressed JPEG images*, IEEE Transactions on Information Forensics and Security, 9 (2014), pp. 1299–1310.

[85] T. GLOE, *Forensic analysis of ordered data structures on the example of JPEG files*, in 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Dec 2012, pp. 139–144.

[86] T. GLOE, M. KIRCHNER, A. WINKLER, AND R. BÖHME, *Can we trust digital image forensics?*, in Proceedings of the 15th ACM international conference on Multimedia, ACM, 2007, pp. 78–86.

[87] T. GLOE AND B. R.:, *The dresden image database for benchmarking digital image forensics*, Journal of Digital Forensic Practice, 3 (2010), pp. 150–159.

[88] M. GOLJAN, M. CHEN, P. COMESAÑA, AND J. FRIDRICH, *Effect of compression on sensor-fingerprint based camera identification*, Electronic Imaging, 2016 (2016), pp. 1–10.

[89] M. GOLJAN, J. FRIDRICH, AND C. M.:, *Defending against fingerprint-copy attack in sensor-based camera identification*, IEEE Transactions on Information Forensics and Security, 6 (2011), pp. 227–236.

[90] M. GOLJAN, J. J. FRIDRICH, AND T. FILLER, *Managing a large database of camera fingerprints.*, Media Forensics and Security, 754108 (2010).

[91] F. H.:, *Exposing digital forgeries from JPEG ghosts*, IEEE Transactions on Information Forensics and Security, 4 (2009), pp. 154–160.

[92] L. H.:, *Countering anti-JPEG compression forensics*, in 19th IEEE International Conference on Image Processing (ICIP, 2012, pp. 241–244.

[93] A. HAOUZIA AND R. NOUMEIR, *Methods for image authentication: a survey*, Multimedia tools and applications, 39 (2008), pp. 1–46.

[94] Z. HE, W. LU, W. SUN, AND H. J., *Digital image splicing detection based on Markov features in DCT and DWT domain*, Pattern Recognition, 45 (2012), pp. 4292–4299.

[95] A. HILL THEODORE P.:, *statistical derivation of the significant-digit law*, Statistical Science, (1995), pp. 354–363.

[96] W. HOU, Z. JI, X. JIN, AND L. X.:, *Double JPEG compression detection base on extended first digit features of DCT coefficients*, International Journal of Information and Education Technology, 3 (2013), pp. 512–515.

[97] Y. HUANG, J. ZHANG, AND H. HUANG, *Camera model identification with unknown models*, IEEE Transactions on Information Forensics and Security, 10 (2015), pp. 2692–2704.

[98] C. A. J.:, *Improved photo response non-uniformity (prnu) based source camera identification*, Forensic science international, 226 (2013), pp. 132–141.

[99] S. D. J.:, *Handbook of parametric and nonparametric statistical procedures*, crc Press, lam, 2000 (2003).

[100] C. J-H., J. W. SHIN, N. S. KIM, AND M. S. K.:, *Image probability distribution based on generalized gamma function*, IEEE Signal Processing Letters, 12 (2005), pp. 325–328.

[101] M. K. JOHNSON AND H. FARID, *Exposing digital forgeries through chromatic aberration*, in Proceedings of the 8th workshop on Multimedia and security, ACM, 2006, pp. 48–55.

[102] I. KARA, *Investigation of ballistic evidence through an automatic image analysis and identification system*, Journal of forensic sciences, 61 (2016), pp. 775–781.

[103] E. KEE AND H. FARID, *Digital image authentication from thumbnails.*, in Media Forensics and Security, 2010, p. 75410E.

[104] ——, *Digital image authentication from thumbnails*, in IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2010, pp. 75410E–75410E.

[105] E. KEE, M. K. JOHNSON, AND H. FARID, *Digital image authentication from jpeg headers*, IEEE transactions on information forensics and security, 6 (2011), pp. 1066–1075.

[106] ——, *Digital image authentication from JPEG headers*, IEEE Transactions on Information Forensics and Security, 6 (2011), pp. 1066–1075.

[107] J. D. KORNBLUM, *Using JPEG quantization tables to identify imagery processed by software*, Digital Investigation, 5, Supplement (2008), pp. S21 – S25.
The Proceedings of the Eighth Annual {DFRWS} Conference.

[108] H. W. KUHN, *The hungarian method for the assignment problem*, Naval Research Logistics Quarterly, 2 (1955), pp. 83–97.

[109] S. LAI AND B. R. C. COUNTER FORENSICS:, *The case of JPEG compression*, Information Hiding, (2011), pp. 285–298.

[110] A. LAM EDMUND Y.:, *mathematical analysis of the DCT coefficient distributions for images*, IEEE Transactions on Image Processing, 9 (2000), pp. 1661–1666.

[111] A. LAWGALY, F. KHELIFI, AND A. BOURIDANE, *Image sharpening for efficient source camera identification based on sensor pattern noise estimation*, in Emerging Security Technologies (EST), 2013 Fourth International Conference on, IEEE, 2013, pp. 113–116.

[112] B. LI, Y. Q. SHI, AND H. J.:, *Detecting doubly compressed JPEG images by using mode based first digit features*, IEEE, 10 (2008), pp. 730–735.

[113] D. LI, *Image processing for the positive identification of forensic ballistics specimens*, in Proceedings of the Sixth International Conference on Information Fusion, Cairns, Queensland, Australia, 2003, pp. 1494–1498.

[114] X. H. LI, Y. Q. ZHAO, M. LIAO, F. Y. SHIH, AND Y. Q. S.:, *Detection of tampered region for jpeg images by using mode-based first digit features*, EURASIP Journal on advances in signal processing, 1 (2012), pp. 1–10.

[115] K.-L. LIM AND H. GALOOGAHI, *Shape classification using local and global features*, in Pacific-Rim Symposium on Image and Video Technology, 2010, pp. 115–120.

[116] Q. LIU, A. H. SUNG, Z. CHEN, AND C. L.:, *Exposing image tampering with the same quantization matrix*, Multimedia Data Mining and Analytics, (2015), pp. 327–343.

[117] J. LUKAS, J. FRIDRICH, AND M. GOLJAN, *Digital camera identification from sensor pattern noise*, IEEE Transactions on Information Forensics and Security, 1 (2006), pp. 205–214.

[118] J. LUKÁŠ AND F. J.:, *Estimation of primary quantization matrix in double compressed JPEG images*, in Proceedings of the Digital Forensic Research Workshop, 2003, pp. 5–8.

[119] W. LUO, Z. QU, J. HUANG, AND G. QIU, *A novel method for detecting cropped and recompressed image block*, in Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 2, IEEE, 2007, pp. II–217.

[120] W. LUO, Z. QU, J. HUANG, AND G. QIU, *A novel method for detecting cropped and recompressed image block*, in IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, April 2007, pp. II.217–220.

[121] S. MANIMURUGAN AND A. JOSE B.:, *Novel method for detecting triple JPEG ompression with the same quantization matrix*, International Journal of Engineering Trends and Technology, 3 (2012), pp. 94–97.

[122] D. MARR, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Henry Holt and Co., Inc., New York, NY, USA, 1982.

[123] G. MCNEILL AND S. VIJAYAKUMAR, *Hierarchical procrustes matching for shape retrieval*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 885–894.

[124] J. MIANO, *Compressed image file formats: jpeg, png, gif, xbm, bmp*, Addison-Wesley Professional, 1999.

[125] S. MILANI, M. TAGLIASECCHI, AND T. S.:, *Antiforensics attacks to benford's law for the detection of double compressed images*, in International Conference on Acoustics Speech and Signal Processing (ICASSP, 2013, pp. 3053–3057.

[126] S. MILANI, M. TAGLIASECCHI, AND T. S., *Discriminating multiple JPEG compressions using first digit features*, APSIPA Transactions on Signal and Information Processing, 3 (2014).

[127] J. S. MINGSI TONG, W. CHU, AND R. M. THOMPSON, *Fired cartridge case identification using optical images and the congruent matching cells (cmc) method*, Journal of research of the National Institute of Standards and Technology, 119 (2014), p. 575.

[128] M. MOLTISANTI, A. PARATORE, S. BATTIATO, AND L. SARAVO, *Image manipulation on facebook for forensics evidence*, in International Conference on Image Analysis and Processing, Springer, 2015, pp. 506–517.

[129] G. MUHAMMAD, M. HUSSAIN, K. KHAWAJI, AND G. BEBIS, *Blind copy move image forgery detection using dyadic undecimated wavelet transform*, in Digital Signal Processing (DSP), 2011 17th International Conference on, IEEE, 2011, pp. 1–6.

[130] S. MUNDER, C. SCHNÖRR, AND D. M. GAVRILA, *Pedestrian detection and tracking using a mixture of view-based shape-texture models*, IEEE Transactions on Intelligent Transportation Systems, 9 (2008), pp. 333–343.

[131] H. MURAKAMI, Y. HATORI, AND Y. H.:, *Comparison between dpcm and hadamard transform coding in the composite coding of the NTSC color TV signal*, IEEE Transactions on Communications, 30 (1982), pp. 469–479.

[132] P. N.:, *Defending against statistical steganalysis*, Usenix Security Symposium, 10 (2001), pp. 323–336.

[133] B. P.:, *â break our steganographic systemâ: The ins and outs of organizing boss, information hiding*, Springer, (2011), pp. 59–70.

[134] X. PAN AND S. LYU, *Detecting image region duplication using sift features*, in Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, IEEE, 2010, pp. 1706–1709.

[135] X. PAN, X. ZHANG, AND S. LYU, *Exposing image forgery with blind noise estimation*, in Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security, ACM, 2011, pp. 15–20.

[136] C. PASQUINI, G. BOATO, AND P.-G. F.:, *Multiple jpeg compression detection by means of benford-fourier coefficients*, IEEE International Workshop on Information Forensics and Security (WIFS), (2014), pp. 113–118.

[137] C. PASQUINI, G. BOATO, AND A. PEREZ-GONZALEZ F.:, *Benford-fourier jpeg compression detector*, in IEEE International Conference on Image Processing (ICIP, 2014, pp. 5322–5326.

[138] F. PENG AND X.-L. WANG, *Digital image forgery forensics by using blur estimation and abnormal hue detection*, in Photonics and Optoelectronic (SOPO), 2010 Symposium on, IEEE, 2010, pp. 1–4.

[139] C. PERON AND L. M. D. ANTI FORENSICS:, *emerging trends in data transformation techniques*, in Proceedings of the 6th Annual Digital Forensic Research Workshop, 2006.

[140] T. PEVNY AND F. J.:, *Detection of double-compression in JPEG images for applications in steganography*, IEEE Transactions on Information Forensics and Security, 3 (2008), pp. 247–258.

[141] A. PIVA, *An overview on image forensics*, ISRN Signal Processing, 2013 (2013), p. 22.

[142] P. W. K. D. I. PROCESSING, J. WILEY, AND N. Y. SONS, *1191*, (1978), pp. 491–556.

[143] G. PUGLISI AND S. BATTIATO, *A robust image alignment algorithm for video stabilization purposes*, IEEE Transactions on Circuits and Systems for Video Technology, 21 (2011), pp. 1390–1400.

[144] A. PURI AND W. A. H.:, *Spatial-domain resolution-scalable video coding*, Visual Communications', 93 (1993), pp. 718–729.

[145] J. QI, F. XIN, L. ZHONGXUAN, L. YU, AND G. HE, *A new geometric descriptor for symbols with affine deformations*, Pattern Recognition Letters, 40 (2014), pp. 128–135.

[146] J. R. QUINLAN, *Induction of decision trees*, Machine learning, 1 (1986), pp. 81–106.

[147] H. R.:, *Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem*, Digital Investigation, 3 (2006), pp. 44–49.

[148] D. RAVÌ, G. FARINELLA, V. TOMASELLI, M. GUARNERA, AND B. S.:, *Representing scenes for real-time context classification on mobile devices*, Pattern Recognition, 48 (2015), p. 4.

[149] K. J. RAY LIU, *Multimedia forensics: Where sherlock holmes meets signal processing*, ICME, (2006).

[150] J. A. REDI, W. TAKTAK, AND J.-L. DUGELAY, *Digital image forensics: a booklet for beginners*, Multimedia Tools and Applications, 51 (2011), pp. 133–162.

[151] J. A. REDI, W. TAKTAK, AND D. J. L. D. IMAGE FORENSICS:, *a booklet for beginners*, Multimedia Tools and Applications, 51 (2011), pp. 133–162.

[152] R. REININGER AND G. J. D.:, *Distributions of the two-dimensional DCT coefficients for images*, IEEE Transactions on Communications, 31 (1983), pp. 835–839.

[153] C. RIESS AND E. ANGELOPOULOU, *Scene illumination as an indicator of image manipulation*, in Information Hiding, Springer, 2010, pp. 66–80.

[154] N. ROMA AND A. SOUSA L.:, *tutorial overview on the properties of the discrete cosine transform for encoded image and video processing*, Signal Processing, 91 (2011), pp. 2443–2464.

[155] K. ROSENFELD AND H. T. SENCAR, *A study of the robustness of prnu-based camera identification*, in IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2009, pp. 72540M–72540M.

[156] M. S.:, *Segmentation-based motion compensation for enhanced video coding*, in Conference on Acoustics Speech and Signal Processing, I. International, ed., mar, 2012, ICASSP), pp. 2253–2256.

[157] K. SAN CHOI, E. Y. LAM, AND K. K. WONG, *Source camera identification using footprints from lens aberration.*, in Digital Photography, 2006, p. 60690J.

[158] G. SCHAEFER AND S. M. UCID:, *an uncompressed color image database, electronic imaging*, International Society for Optics and Photonics, (2003), pp. 472–480.

[159] B. SHIVAKUMAR AND S. BABOO, *Detection of region duplication forgery in digital images using surf*, International Journal of computer science Issues, 8 (2011), pp. 199–205.

[160] M. S. SREELAKSHMI AND V. D.:, *Image compression using anti-forensics method, international journal of computer science*, Engineering & Applications, 3 (2013), p. 1.

[161] M. C. STAMM, W. S. LIN, AND L. K. J.:, *Temporal forensics and anti-forensics for motion compensated video*, IEEE Transactions on Information Forensics and Security, 7 (2012), pp. 1315–1329.

[162] M. C. STAMM, M. WU, AND K. J. R. LIU, *Information forensics: An overview of the first decade*, IEEE Access, 1 (2013), pp. 167–200.

[163] A. SWAMINATHAN, M. WU, AND K. R. LIU, *Non-intrusive forensic analysis of visual sensors using output images*, in Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, vol. 5, IEEE, 2006, pp. V–V.

[164] D. T. DANG-N. AL.: RAISE, *â a raw images dataset for digital image forensics*, ACM Multimedia Systems, Portland, Oregon, March, (2015), pp. 18–20.

[165] A. TORRALBA AND E. A. A.:, *Unbiased look at dataset bias*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 2011, pp. 1521–1528.

[166] G. VALENZISE, M. TAGLIASACCHI, AND T. S.:, *Revealing the traces of JPEG compression anti-forensics*, IEEE Transactions on Information Forensics and Security, 8 (2013), pp. 335–349.

[167] D. VALSESIA, G. COLUCCIA, T. BIANCHI, AND E. MAGLI, *Compressed fingerprint matching and camera identification via random projections*, IEEE Transactions on Information Forensics and Security, 10 (2015), pp. 1472–1485.

[168] 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, IEEE, 2007, pp. 883–886.

[169] S. VELASCO-FORERO AND J. ANGULO, *Statistical shape modeling using morphological representations*, in International Conference on Pattern Recognition, 2010, pp. 3537–3540.

[170] L. W.:, *JPEG error analysis and its applications to digital image forensics*, IEEE Transactions on Information Forensics and Security, 5 (2010), pp. 480–491.

[171] S. WALTON, *Image authentication for a slippery new age*, Dr. Dobb's Journal, 20 (1995), pp. 18–26.

[172] B. WANG, W. SHEN, W.-Y. LIU, X.-G. YOU, AND X. BAI, *Shape classification using tree-unions*, in International Conference on Pattern Recognition, 2010, pp. 983–986.

[173] J. WANG, X. BAI, X. YOU, W. LIU, AND L. LATECKI, *Shape matching and classification using height functions*, Pattern Recognition Letters, 33 (2011), pp. 134–143.

[174] M. WANG, Z. CHEN, W. FAN, AND X. Z.:, *Countering anti-forensics to wavelet-based compression*, in IEEE International Conference on Image Processing (ICIP, 2014, pp. 5382–5386.

[175] W. WANG, J. DONG, AND T. T.:, *Exploring DCT coefficient quantization effects for local tampering detection*, IEEE Transactions on Information Forensics and Security, 9 (2014), pp. 1653–1666.

[176] A. WESTFELD AND P. A.:, *High capacity despite better steganalysis (f5–a steganographic algorithm), information hiding*, 4 (2001), pp. 289–302.

[177] B. WIDROW, I. KOLLAR, AND L. M.-C.:, *Statistical theory of quantization*, IEEE Transactions on Instrumentation and Measurement, 45 (1996), pp. 353–361.

[178] L. E. Y.:, *Analysis of the DCT coefficient distributions for document coding*, IEEE Signal Processing Letters, 11 (2004), pp. 97–100.

[179] C. Y.-L. AND H. C.-T.:, *Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection*, IEEE Transactions on Information Forensics and Security, 6 (2011), pp. 396–406.

[180] J. YANG, G. ZHU, J. HUANG, AND Z. X.:, *Estimating JPEG compression history of bitmaps based on factor histogram*, Digital Signal Processing, 2015.

[181] J. YANG, G. ZHUA, AND H. J.:, *Detecting Doubly Compressed JPEG Images by Factor Histogram*, Asia Pacific Signal and Information Processing Association - Annual Summit and Conference (APSIPA ASC, 2011.

[182] L. Z.:, *First JPEG Quantization Matrix Estimation Based on Histogram Analysis*, Pattern Recognition, 2013.

[183] F. ZACH, C. RIESS, AND A. E.:, *Automated image forgery detection through classification of JPEG ghosts*, Springer, 2012.

[184] M. D. ZEILER AND R. FERGUS, *Visualizing and understanding convolutional networks*, in European conference on computer vision, Springer, 2014, pp. 818–833.

[185] M. D. ZEILER, D. KRISHNAN, G. W. TAYLOR, AND R. FERGUS, *Deconvolutional networks*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2528–2535.

[186] D. ZHANG AND G. LU, *Review of shape representation and description techniques*, Pattern Recognition, 37 (2004), pp. 1–19.

[187] X. ZHAO, J. LI, S. LI, AND S. WANG, *Detecting digital image splicing in chroma spaces*, in International Workshop on Digital Watermarking, Springer, 2010, pp. 12–22.