University of Catania

Department of Mathematics and Computer Science

XXXIII Ph.D. in Computer Science (International)

Francesco Guarnera

Advanced Methods for Image Forensics: First Quantization Estimation and Document Authentication

Doctoral Thesis

Advisor: Prof. Sebastiano Battiato

Co-Advisor: Prof. Giovanni Puglisi, Ing. Oliver Giudice

Cycle XXXIV (2018-2021)

“We can live a wonderful life in the world if we know how to work and love: work for those we love and love what we work for. ”

Lev Tolstoj

Dedicated to Eleonora

Acknowledgements

I would like to express my gratitude to the Prof. Sebastiano Battiato. He was a guide during these years, in both research and real life, supporting me to improve my skills and teaching me the professionalism in this field.

A special thank to iCTLab team: Ing. Oliver Giudice for supporting my research activities and for his suggestions, Dr. Antonino Barbaro Paratore, Elisa Virgillito, Serena Patanè, Chiara Maugeri, and all the guys known inside iCTLab over these years for moments (and lunches) shared together. A special mention is for Dr. Roberto Leotta and for my doctoral colleague Luca Guarnera: you were my younger brothers. I will never forget the long talks with Roberto and the London days with Luca.

I would like to thank Prof. Giovanni Puglisi of University of Cagliari; we never met but he managed to pass on his teachings to me with rare professionalism.

Thanks to Prof. Salvatore Livatino for the opportunity to spend some months to the School of Engineering and Technology, University of Hertfordshire, Hatfield, United Kingdom; it was a great opportunity for me.

Thanks to the "Reparto Investigazioni Scientifiche dell'Arma dei Carabinieri" (RIS), of Messina (Italy); they gave me the opportunity to acquire technical, practical and scientific knowledge in Forensic Science and to use sophisticated devices to conduct my research.

Thanks to all my family and friends, simply for always being there and supporting me.

The biggest thanks go to my wife Viviana. She was the key for this result, with her suggestions and support over these years. She has always believed in me, exhorting me every day to give the best of myself, and guiding me through the difficult choices of life.

Abstract

Digital Forensics, and in a specific way Multimedia Forensics, has grown significantly in the last years. Digital Forensics is defined as the branch of Forensic Science which scientifically analyzes a digital evidence in order to obtain information about it. The Digital 2021 Global Overview Report [1] published in January 2021 certifies that the world's population has reached the number of 7.83 billion. The 66.6% (i.e. 5.22 billion) use a mobile phone, the 59.5% (i.e. 4.66 billion) use internet and the 53.6% (i.e. 4.2 billion) are social media users. The same report declares that from 2015 to 2020 the daily time spent with social media increased of 34 minutes (it was 1 hours and 51 minutes in 2015) and that it's destined to rise. In the described scenario the number of shared images, video and audio (or Multimedia) contents become difficult to manage. The described numbers and types of digital evidences have led to the birth of several fields of Digital Forensics, faced from different communities: Multimedia Security, Computer Forensics and Signal Processing. Image Forensics has the goal to obtain information about the most popular digital evidence: images. Developing new algorithms for forensic purposes was the main focus of my Ph.D.. In this thesis some advanced methods will be presented about two specific tasks: the first one is related to the Camera Model Identification (CMI) with the goal to identify the quantization table employed during the first JPEG compression; the second one exploits the image as the digitization of a real paper sheet in order to extract a unique fingerprint. Both the tasks produced relevant methods, widely compared with state-of-the-art to demonstrate their scientific goodness.

[1] https://wearesocial.com/blog/2021/01/digital-2021-the-latest-insights-into-the-state-of-digital

Publications

Journals

1. Sebastiano Battiato, Oliver Giudice, Francesco Guarnera, Giovanni Puglisi. *First Quantization Estimation by a Robust Data Exploitation Strategy of DCT Coefficients.* IEEE Access Vol. 9, pp. 73110-73120, 2021.

2. Sebastiano Battiato, Oliver Giudice, Francesco Guarnera, Giovanni Puglisi. *Estimating Previous Quantization Factors on Multiple JPEG Compressed Images.* SpringerOpen EURASIP Journal on Information Security Vol. 2021, Num. 1, pp. 1-11, 2021.

3. Francesco Guarnera, Oliver Giudice, Dario Allegra, Filippo Stanco, Sebastiano Battiato, Salvatore Livatino, Vito Matranga, Angelo Salici. *A Robust Document Identification Framework through f-BP Fingerprint.* MDPI Journal of Imaging Vol. 7, Num. 8, pp. 126, 2021.

Journal To appear

1. Sebastiano Battiato, Oliver Giudice, Francesco Guarnera, Giovanni Puglisi. *CNN-based Estimation of the First Quantization Matrix from Double Compressed JPEG Images.* Journal of Visual Communication and Image Representation, 2021.

Conferences

1. Sebastiano Battiato, Oliver Giudice, Francesco Guarnera, Giovanni Puglisi. *Computational data analysis for first quantization estimation on JPEG double compressed images.* IEEE 25th International Conference on Pattern Recognition (ICPR), pp. 5951-5958, 2021.

2. Sebastiano Battiato, Oliver Giudice, Francesco Guarnera, Giovanni Puglisi. *In-depth DCT coefficient distribution analysis for first quantization Estimation.* Springer International Publishing Pattern Recognition, ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI, pp. 573-587, 2021.

3. Sebastiano Battiato, Francesco Guarnera, Alessandro Ortis, Francesca Trenta, Lorenzo Ascari, Consolata Siniscalco, Tommaso De Gregorio, Eloy Suárez. *Pollen Grain Classification Challenge 2020.* Springer, International Conference on Pattern Recognition (ICPR), pp. 469-479, 2021.

4. Oliver Giudice, Dario Allegra, Francesco Guarnera, Filippo Stanco, Sebastiano Battiato. *Animated GIF optimization by adaptive color local table management.* IEEE International Conference on Image Processing (ICIP), pp. 843-847, 2020.

5. Oliver Giudice, Francesco Guarnera, Antonino Paratore, Sebastiano Battiato. *1-D DCT domain analysis for JPEG double compression detection.* Springer International Conference on Image Analysis and Processing (ICIAP), pp. 716-726, 2019.

6. Francesco Guarnera, Dario Allegra, Oliver Giudice, Filippo Stanco, Sebastiano Battiato. *A new study on wood fibers textures: documents authentication through LBP fingerprint.* IEEE International Conference on Image Processing (ICIP), pp. 4594-4598, 2019.

Contents

Abstract **iv**

Publications **v**

1 The Forensic Science **1**

1.1 Digital Forensics 1

1.2 Image Forensics 4

1.2.1 Acquisition 5

1.2.2 Compression 8

1.2.3 Editing and upload 9

2 First Quantization Estimation **12**

2.1 Introduction 12

2.1.1 JPEG 12

2.1.2 JPEG Notation 15

2.2 State of the art 16

2.3 FQE through a JPEG simulation 18

2.3.1 Experimental results 23

2.3.2 Experiments with Multiple Compressions 32

2.4 FQE by a smart data exploitation 37

2.4.1 K-nearest neighbors (KNN) method 37

Parameter setting 43

Experimental Results 45

Deepening 51

2.4.2 Convolution Neural Network (CNN) Method 54

Experimental Results 65

2.5 Summary 72

3 Document Identification 73

3.1 Introduction . . . 73

3.2 State of the art . . . 74

3.3 Printed Document Identification . . . 75

3.3.1 Fingerprint Extraction Process . . . 76

Document Digitization and Image Registration Considerations 77

Extracting a Unique Fingerprint . . . 78

3.3.2 Datasets for Document Identification and Fingerprint Testing 79

Devices . . . 79

Dataset Acquisition . . . 80

Image Registration . . . 81

3.3.3 Experimental results . . . 82

Dataset Comparison . . . 83

Comparisons among LBP Variants . . . 83

Tests on Noisy Environment . . . 84

Fingerprint Dimensions Optimization . . . 85

3.3.4 Fingerprint Robustness Analysis . . . 88

3.4 Summary . . . 91

4 Conclusion 94

Bibliography 97

Chapter 1

The Forensic Science

The Forensic Science (FS) is the application of scientific methods and techniques to matters under investigation by a court of law. The fields of FS are various: Forensic Archaeology, Forensic Pathology, Forensic Geology, Forensic Odontology, Forensic Toxicology, Criminalistics, and Digital Forensics (including Multimedia Forensics). Each of these fields exploits the related scientific methods in order to obtain useful information during an investigation or to be presented in a court.

My Ph.D. was focused in particular on Image Forensics, a specific branch of DF that have the goal to analyze the most common digital evidence: the image. In particular, in the following chapters, the image will be exploited and treated in two different ways: in the Chapter 2 the traces leave inside it by the compression phase will be analyzed in order to reconstruct the image history and to recognize the parameters employed during the compression, while in Chapter 3 the image will be exploited as a digital representation of a real paper sheet in order extract a fingerprint from it.

1.1 Digital Forensics

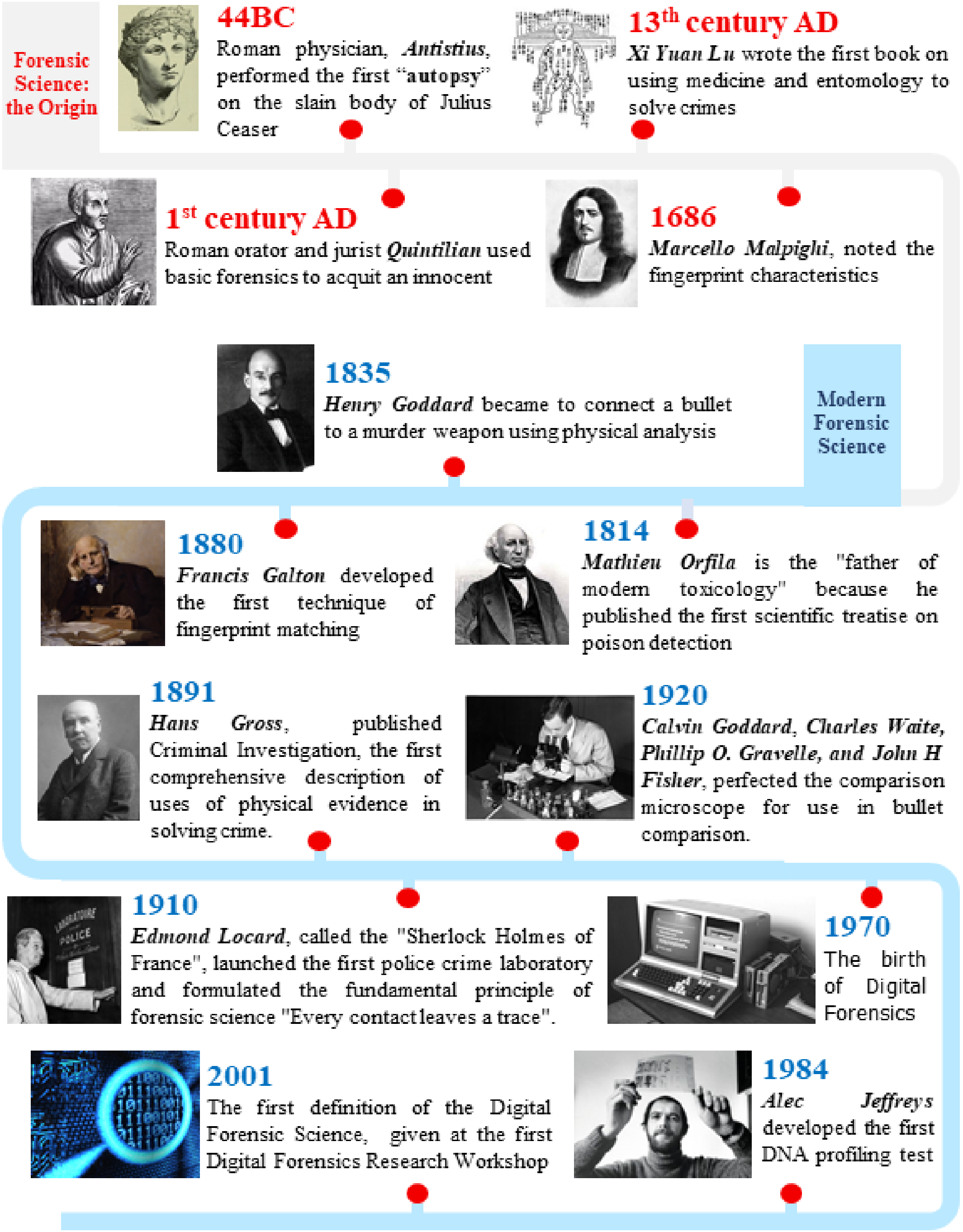
The spread of social networks, with more than 1 billion of images posted every day, has exponentially expanded the problems related to the employing of fake media, that often causes serious repercussions on people's lives, political problems and misinformation. Digital Forensics (DF) was born to oppose this type of issues, obtaining information about the media under analysis and reconstructing its history.

The first notion of DF dates back to 1970s, when in the Federal Rules (US) the importance of digital evidence was discussed. The real digital forensics investigations started in the last part of 1980s, when federal agents began to analyze the digital evidence found inside computers. Non-standardized methods were employed until the 2000s when the importance of these type of evidences was understood by researchers, agents and lawyers; the first research groups in Forensic Science started around 2000 and 2001. A brief overview of the history of Forensic Science can be found in Figure 1.1.

In front of a digital evidence, in general content analysis is the main task. A definition given by a related FBI working group to assess the role of Forensics image/video analysis is: *"Forensics Image (Video) analysis is the application of IMAGE SCIENCE and DOMAIN EXPERTISE to interpret the content of an image or the image itself in legal matters"*.

Digital Forensic Science was defined for the first time in 2001 at the first Digital Forensics Research Workshop (DFRWS): *"The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations"*.

Digital Forensics and its branch of Multimedia Forensics have evolved over time with the birth of specific areas. The main areas of Multimedia Forensics are classified based on their signal type: image, audio and video forensics. In addition to the history reconstruction Multimedia Forensics has the objectives to interpret and analyze the content, to identify the source, to assess integrity and authenticity and also to enhance the media under analysis. Big data and the increasing of digital evidences has opened great challenges for forensic researchers in all the fields, many of which are still open. In the context of image authentication, for example, the European Network of Forensic Science Institutes (ENFSI) has only recently defined the best practices for this task which are constantly updated over the time; the issue derives from the continuous development of increasingly sophisticated manipulation tools and it needs a general agreement to define the evidence analysis standard for this specific task.



Forensic Science: the Origin

44BC
Roman physician, ***Antistius***, performed the first "**autopsy**" on the slain body of Julius Ceaser

13th century AD
Xi Yuan Lu wrote the first book on using medicine and entomology to solve crimes

1st century AD
Roman orator and jurist ***Quintilian*** used basic forensics to acquit an innocent

1686
Marcello Malpighi, noted the fingerprint characteristics

1835
Henry Goddard became to connect a bullet to a murder weapon using physical analysis

Modern Forensic Science

1880
Francis Galton developed the first technique of fingerprint matching

1814
Mathieu Orfila is the "father of modern toxicology" because he published the first scientific treatise on poison detection

1891
Hans Gross, published Criminal Investigation, the first comprehensive description of uses of physical evidence in solving crime.

1920
Calvin Goddard, Charles Waite, Phillip O. Gravelle, and John H Fisher, perfected the comparison microscope for use in bullet comparison.

1910
Edmond Locard, called the "Sherlock Holmes of France", launched the first police crime laboratory and formulated the fundamental principle of forensic science "Every contact leaves a trace".

1970
The birth of Digital Forensics

2001
The first definition of the Digital Forensic Science, given at the first Digital Forensics Research Workshop

1984
Alec Jeffreys developed the first DNA profiling test

Figure 1.1: Brief overview of the history of Forensic Science.

In this Ph.D. thesis, I will mainly focus on Image Forensics. Images are the most common type of evidence; methods able to obtain information about the history of them will be analyzed and presented. Every image contains traces about the process that has generated it or edited it, then the objective is to find those traces and use them to reconstruct the process from the original source to the evidence.

1.2 Image Forensics

The use of images has always represented the most immediate and understandable way to communicate. Since the beginning of the third millennium the rapid diffusion of smartphones allowed any person to own a device which permits the acquisition of visual data. In addiction to that, the availability of editing tools and the possibility to share these visual contents with anyone has made images the most widespread but at the same time the most counterfeited media. Forensics Science has the goal to exploit evidences to facilitate the reconstruction of criminal events. Today, the ability to access this large number of digital evidences represents a great opportunity but, at the same time, a great challenge, deriving from the difficulty to validate an evidence: phenomenon like Deepfake (i.e.fake images generated with the help of artificial intelligence) are the enemies to beat.

The reconstruction of the image's history [1] is an open task from about thirty years but, the recent exponential usage of images required more and more advanced methods. Today, as described in Fig. 1.2, a typical image lifecycle concerns the following steps (in most cases):

1. Acquisition of the image through a digital camera
2. Editing of the image
3. Upload of the image to a Social Network

Every step includes a number of actions that, in many cases, could be analyzed in order to reconstruct the image history or more generally, to extract information usable in court. Many methods able to obtain image's history information have been proposed in the years and their first classification is based on the source. The "active" methods add some information in the source image so that it is possible to analyze the information added previously to understand possible editing of the image. Example of active methods are based on trustworthy cameras [2], watermarking [3] or crypted digital signature. The "passive" methods are based instead, on a blind analysis of the evidence without any a-priori knowledge; in these methods the traces left during the image generation phases, from acquisition to editing, are analyzed. It is easy to understand that passive methods have an higher level of complexity. In addition, the number of evidences without previously added information represents the most of the digital evidences in digital investigations.

In the following sections three important steps of image's lifecycle will be analyzed and described; furthermore, in order to understand how the investigation on the image is conducted the most advanced methods will be showed and compared.

1.2.1 Acquisition

Fig. 1.3 describes the step of acquisition. This phase could be performed through different devices, but in general the steps of in-camera processing shown in Fig. 1.3 are constant: the image is firstly captured, the undesired light spectrum is minimized through the use of optical filters and the remaining light is focused by the lenses on the sensor. Before reaching the sensor the light is often filtered by the Color Filter Array (CFA) that extract the red-green-blue (RGB) components and select the right light to employ for the components inside the sensor. The output of the sensor is then interpolated in order to obtain a digital color for each pixel through a so-called demosaicing process. The raw digital image in output go through an internal camera process of enhancement which includes color balancing and contrast and gamma correction and finally the compression is performed.

The aforementioned process could leave traces in every step, that in most cases, cannot be perceived by human eyes. The manufacturing process of the lens creates an optical system which is almost unique due to the inperfections on lighting focus; this small lens distortion produces types of aberration that in some cases could be analyzed for forensics purposes; Johnson and Farid in [4] for example exploit lateral chromatic aberration for tampering detection. The CFA represents also a traces leaver: each sensor element captures light inside certain wavelenghts range, introducing a pattern that, if perturbed as part of a manipulation, it appears anomalous. Popescu and Farid [5] in 2005 analyzed the periodic correlation of CFA patterns with a linear model to distinguish natural images from artificial ones. In Fourier domain the natural images present a periodic strong peak, which is absent in the artificial ones. Ferrara et al. in [6] analyzed the same problem with Bayesian framework in

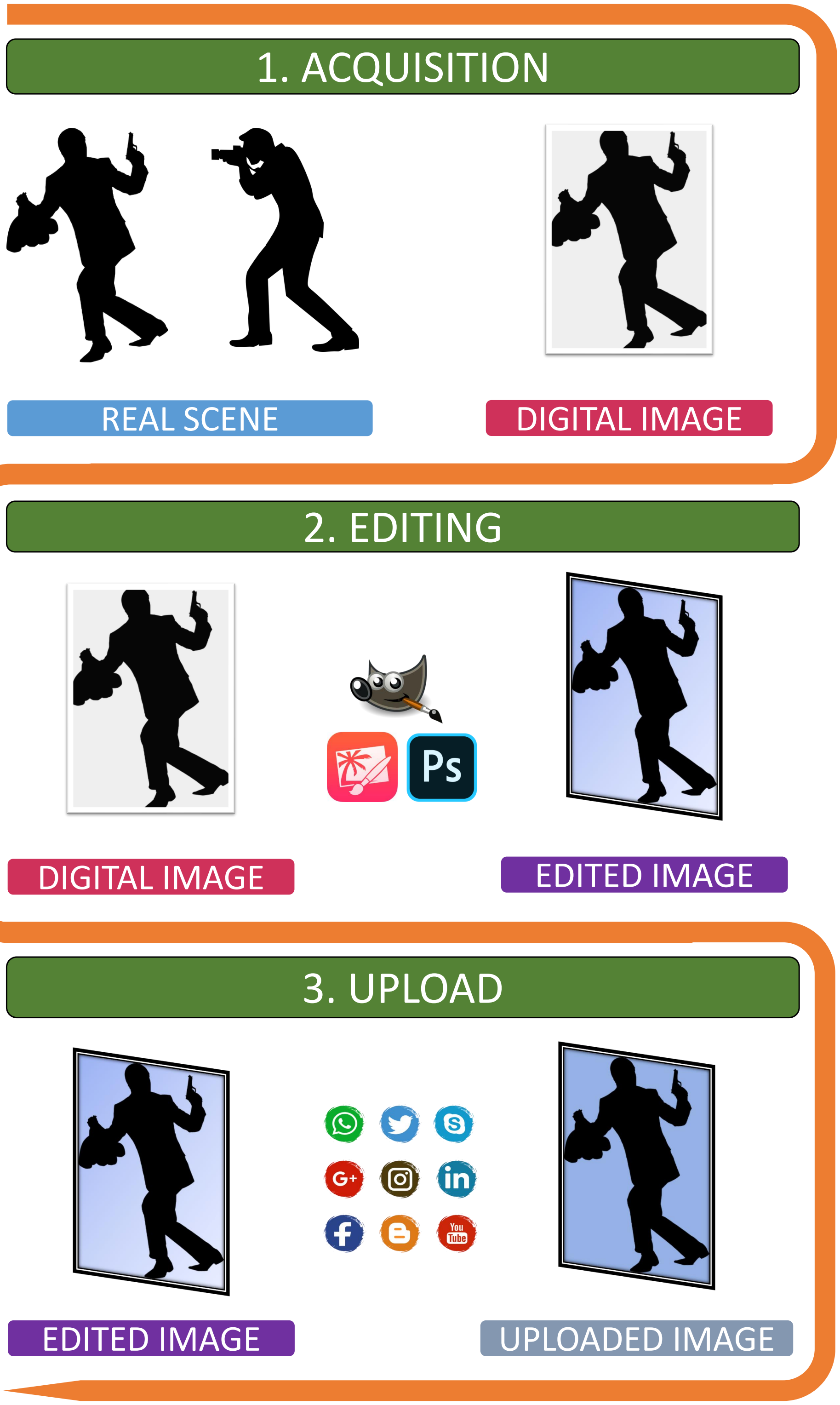


Figure 1.2: A typical image lifecycle.

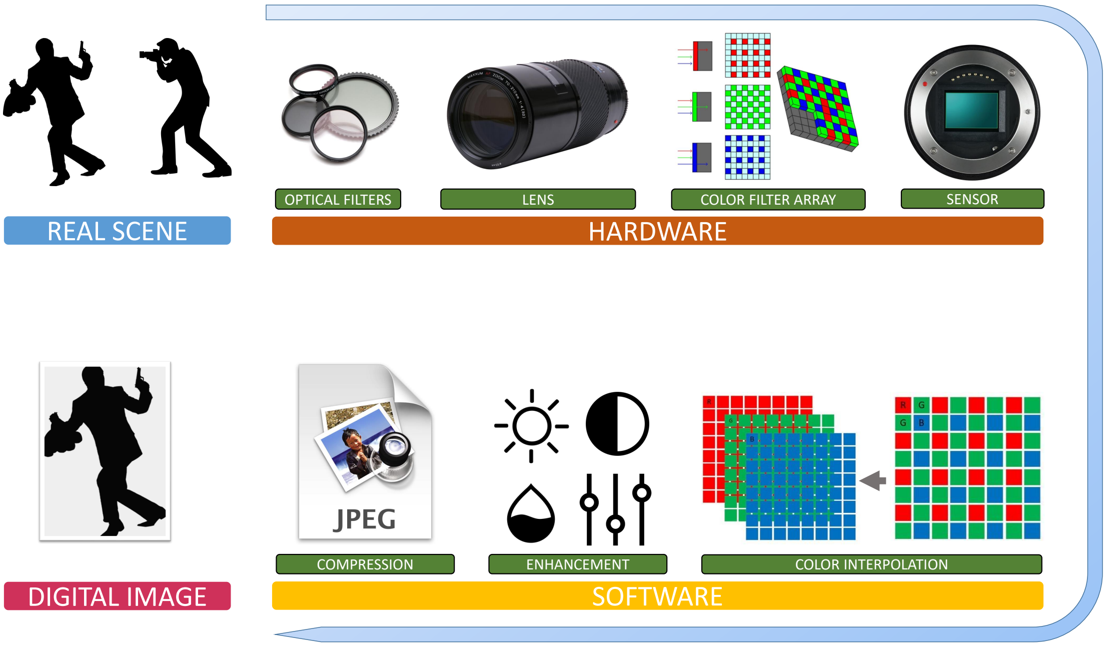


Figure 1.3: Technical image lifecycle.

order to obtain a tampering probability map. Another possible artifact introduced in the acquisition phase is due to the small deviations of the sensor that produce a specific pattern noise called photo-response non-uniformity (PRNU), as explained in [7]. Each image acquired by the same camera contains traces left by the PRNU, now considered as a camera fingerprint because it remains generally stable in camera's lifetime. PRNU is exploited for both Camera Source Identification (CSI) and Tampering Detection because being a fingerprint of the camera, it recognizes the camera itself but also any variations in the fingerprint. PRNU-based forgery detection was proposed in [8] for the first time through a comparison between camera PRNU (obtained before) and the PRNU of target image. Successive PRNU-based works focused on improving the PRNU estimation model ([9, 10]).

Camera design choices could be employed as model local features for Camera Model Identification (CMI); when cameras of the same model share properties of both hardware and software, then it is possible to extract local descriptors to build a statistical model and discriminate device types [11].

1.2.2 Compression

Although the compression step is a part of acquisition (Fig. 1.3) it must be deeply analized because it represents the most obvious source of traces. Lossy image compression is the most common, due to the allowed space saving, which is also the reason why the first compression is usually performed inside the camera. JPEG (Joint Photographic Experts Group) [12] is the most popular and employed compression alghoritm, just imagine that almost all social networks perform a JPEG compression during image upload (deep details about the pipeline of JPEG compression will be shown in Chapter 2). JPEG compression is performed on three basic steps:

- division in 8×8 blocks and transition from spatial domain to frequency domain, through the use of Discrete Cosine Transform (DCT)
- quantization of 8×8 blocks in frequency domain through a quantization matrix
- coding of the data

The exploitation of compression artifacts, mostly on JPEG, represents the bigger area of proposed methods for forensic purposes. Some of them are available in spatial domain like the methods based on the exploitation of the block artifact grid (BAG): the authors of [13, 14], for example, exploited the discontinuities along the block boundaries of JPEG compressed images in order to identify the compression history, also with low level of compression. Bruna et al. in [15] instead, propose a new method to detect cropped images by analyzing the blocking artifacts produced by a previous JPEG compression. Another line of research is based on the analysis of data in the frequency domain. Luo et al. in [16], observing the integral of DCT coefficient histograms, detect previous JPEG compression; in [17] and [18] the authors analyzed the statistical distribution of DCT coefficients comparing them with the Benford law [19], obtaining information about the introduction of compression artifacts; in [20] H. Farid detects manipulated areas through the so-called JPEG ghosts when two JPEG compressions with the same quality factor were performed.

In order to recover information about the acquisition device [21, 22, 23] the forensic analysis of JPEG images specializes to several different tasks. One of them is the Double Quantization Detection (DQD) which has the objective to detect if an image has been JPEG compressed at least twice in order to obtain information about the authenticity of the image or about possible tampering. The Forensic community has been spending a lot of effort on DQD, facing the problem in different scenarios. The first classification is about aligned and not-aligned JPEG double compression: the aligned DQD applies the second compression on the same blocks while in the not-aligned DQD the grid of 8×8 block is offset with respect to the first compression.

Among the aligned DQD methods some are based on natural distribution of first digit (Benford law) as [24], other on DCT application variant [25] and other on periodicity of DCT histograms [26]. The analysis of DQD moved often the research to the tampering detection [27, 28]. Among not-aligned DQD in [29] the blocking artifacts are investigated again with the use of a classifier while [30, 31] employ a simple treshold detector. Some methods manage to locate both aligned and not-aligned DQD: in [32] the authors proposed the detection combining features of spatial and frequency domain while in [33] Bianchi and Piva proposed a statistical model able to individuate the tampering regions in both the aligned and not-aligned scenarios. This task, also with the research of multiple compressions was treated both in a forensics and a counterforensics point of view ([34, 35]).

Recently, machine learning (ML) and deep learning (DL) based methods have revolutionized this task, individuating the right features to employ directly from the data. ML and DL based approaches [36, 37] reach good performances also in presence of difficult scenarios with respect to statistical approaches, but as it often happens, the models produced suffer heavily of overfitting.

In a digital forensics investigation, the DQD is often followed by the First Quantization Estimation FQE, which in some cases is carried out together [38]. FQE has the goal to estimate the matrix employed in the first JPEG compression in order to obtain information about the camera model. The FQE task was the main task of this thesis, for this reason, it will be deeply explored in chapter 2.

1.2.3 Editing and upload

The manipulation of an image often produces traces, also related to re-compression. Basic transformations like rotation, scaling, blurring applied to an image could improve its quality or in maliciously change its semantic content. Some methods focus on the anomalies of the transformations [39, 40] in order to detect a manipulation.

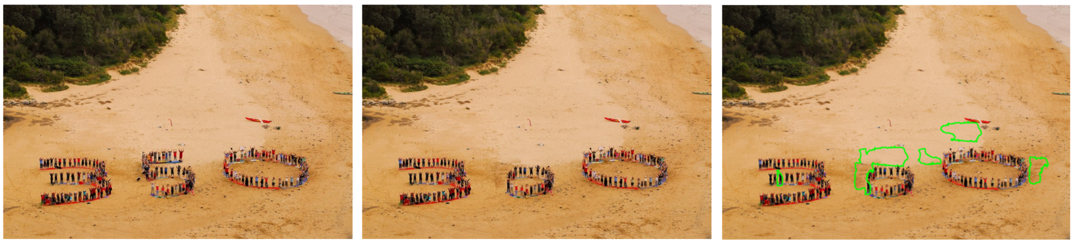


Figure 1.4: Example of copy move detection. From left to right: original image, image after copy move and detection.

Other type of image editing are executed with a malicious intent; copy-move and its derivatives (cut and paste, copy and paste) belong to this class. Copy-move consists in copying a part of an image and past it in another position in order to hide or duplicate content. The copy-move detection (figure 1.4) is the more difficult tampering detection because, being the copied part from the same image, some components are compatible with others and then the forensics methods based on incompatibility are useless; efficient solutions for this type of tampering are based on keypoints [41] or dense-field [42] matching.

Machine learning and deep learning based methods have taken over [43, 44]. The possibility to obtain the best suitable features in an automatic way to discriminate and localize a tampering made previous techniques obsolete. At the same time, antiforensics attacks exploited the potential of machine learning to improve the tampering techniques, hiding the traces used by detection methods and making the task increasingly complicated to deal with.

In the last described scenario, a phenomenon to underline is the so-called Deepfake. They can be defined as fake images generated with the help of deep learning tools like autoencoders (AE) or generative adversarial networks (GAN). Naturally, deepfake could be employed for malicious purposes making the recognition of an original source more difficult. An overview on Media forensics with particular focus on Deepfakes has been proposed in [45]. Several State of the art approaches demonstrated that Deepfake images contain a pattern left by generative models (through convolution layers) that characterizes that specific deep neural architecture [46, 47, 48]. To capture this trace, Guarnera et al.[46, 47] used the Expectation-Maximization Algorithm obtaining features able to distinguish real images from

Deepfake ones. Giudice et al.[48] demonstrated that the so-called GAN Specific Frequencies (GSF), a pattern that characterize several GAN engines can be extracted through the DCT. Jain et al. [49] proposed a framework composed by three levels able to solve several classification tasks: Original Vs Altered (first level); Retouched Vs GAN generated (second level); image classifications created by four GAN engines (third level).

An important phase of image lifecycle has become the Social Network upload; every social platform defines its upload rules, that in time could change. Although in most cases that phase is a composition of others (crop, resizing, compression) conducting a specific analysis for the upload phase can be useful. Some works in the literature strictly analyze this phase [50] building an engine for the image's reconstruction history and detecting not only the compression parameters but also identifying the Social Network.

Chapter 2

First Quantization Estimation

2.1 Introduction

As explained in chapter 1, if a previously JPEG compression in the image under analysis has been detected the following step is the so-called First Quantization Estimation (FQE). It has the goal to estimate the parameters of the previous compression and in particular the quantization matrix employed. The knowledge of the matrix permits to obtain information about the camera model for forensic purposes.

In this chapter the most relevant methods of the state-of-the-art will be present, in order to understand how the FQE task was faced over the years. Subsequently, three different methods will be presented (in the aligned and not scaled scenario) and compared with the aforementioned methods; the presented methods overcome the results w.r.t. the state-of-the-art ones and moreover they achieve surprising results working with images compressed with both standard and custom quantization matrices which represents a relevant property in this research field.

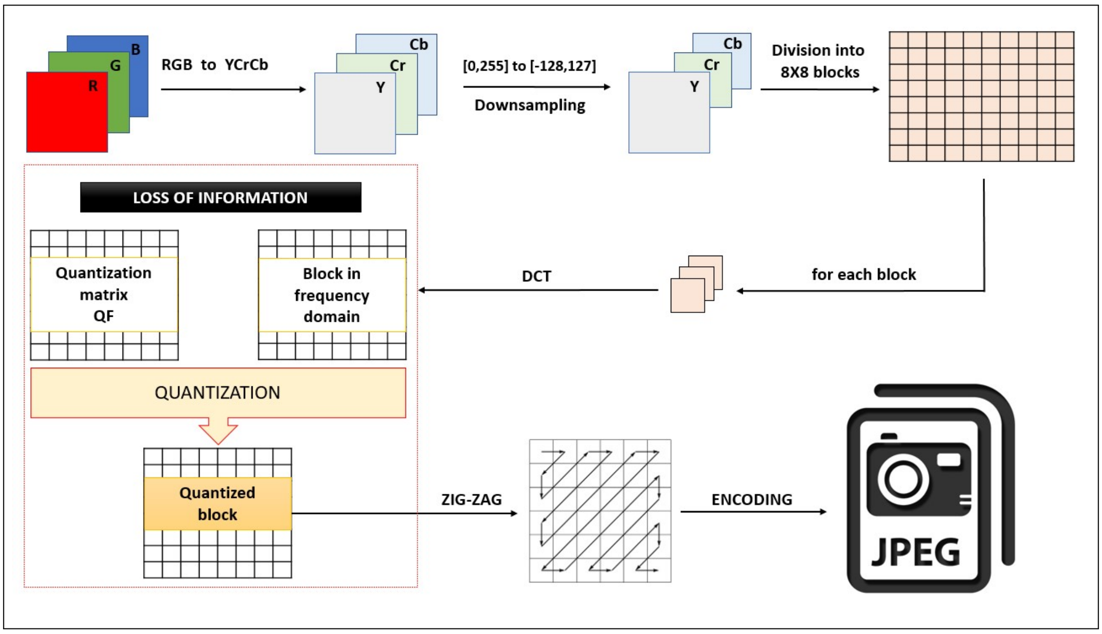
2.1.1 JPEG

The predominant use of JPEG compression and the importance of Social Networks as a source of trial evidence has led forensic research to focus on the analysis of images compressed with this format. In order to understand how the forensic analysis is carried out on JPEG images, the compression and decompression phases of the algorithm will be described below. The primary goal of compression is to reduce the data size. Many formats simply encode the data (by reducing the size) allowing to go back to the original image exactly as it was before compression; this type of compression is called lossless. JPEG is a lossy compression (with loss of information):

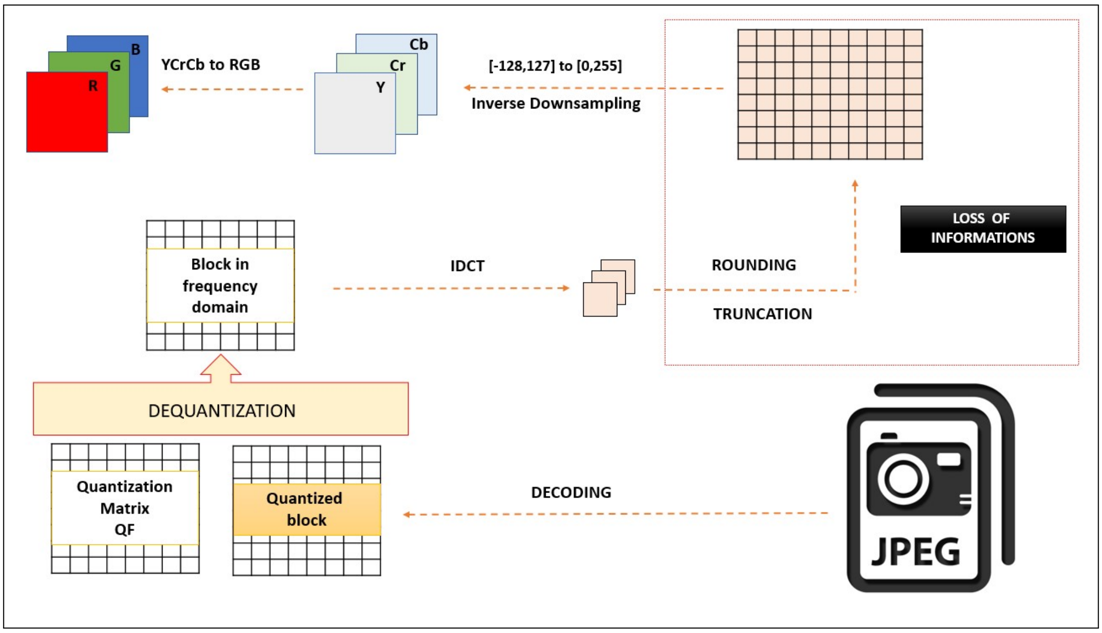
the amount of information lost during compression is delegated directly to whoever performs the compression. In fact, JPEG uses mechanisms by which it is possible to decide how much information to lose at the expense of the quality of the image itself. Figure 2.1a shows the compression phase.

The image, initially on the RGB color space with values in the range [0,255], is switched to the YCrCb color space in the range [-128,127]; this step is done because the small variations of light are perceived more by the human eye and for the same reason in the next step a sub-sampling of the color channels is done. Subsequently, each of the image channels is divided into 8x8 blocks and the Discrete Cosine Transform (DCT) is applied to each of them, which allows to pass from the spatial to the frequency domain. The purpose of this step is to have the frequencies ordered. The quantization phase is the part of the algorithm with the irreversible loss of information. Each of the 8×8 blocks is divided by a quantization matrix and the decimal values are rounded to integer values; generally it provides a stronger quantization (or in many cases complete elimination) of the high frequencies and a lower one on the low frequencies, due to the higher sensitivity of the human eye towards the latter. This generates the first irreversible data loss, as it will no longer be possible to go back to the original value. At this point, the quantized blocks are organized (in zig-zag order) and encoded, creating the compressed file. Figure 2.1b shows the decompression phase.

The compressed image is first decoded and the 8×8 blocks are de-quantized (i.e. multiplied by the quantization matrix saved in the image metadata), reconstructing the 8×8 coefficients blocks in the frequency domain which, of course, will have different values from those preceding the quantization (quantization error). The 8×8 blocks are then subjected to the Inverse Discrete Cosine Transform (IDCT) for the transition to the spatial domain; this step generates two other errors: the decimal values after the IDCT are rounded to integers (rounding error) and furthermore if there are values outside the range [-128,127] these are clipped to the nearest limit (truncation error). Finally, with the inverse subsampling and the switch of color space from YCrCb to RGB we obtain the image display. As can be guessed, the amount of information lost depends entirely on the quantization matrix employed: an identity quantization matrix with value all equal to 1 has no loss of information, but neither does it reduce space. The number of 8×8 quantization matrices is



(a) JPEG compression



(b) JPEG decompression

Figure 2.1: JPEG compression (a) and decompression (b) with all steps.

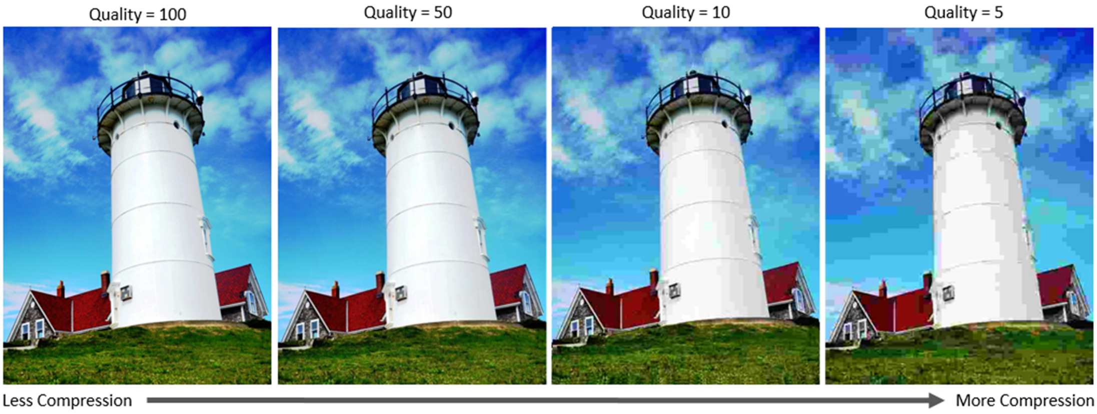


Figure 2.2: JPEG file at varying Quality Factors.

enormous but the JPEG working group has defined a formula to derive 100 standard tables with a quality level associated, defined as Quality Factor (QF). The Quality Factor 100 identifies an identical matrix with values 1 which does not lose information but which does not reduce space. The standard tables represent only indications provided by the JPEG working group; many digital camera suppliers, for example, define customized compression matrices, which, as we will see later, is fundamental for forensic analysis. Figure 2.2 shows different views of the same image compressed with different QF.

2.1.2 JPEG Notation

In this Section the notation employed in all the chapter is defined. Given a raw image I, JPEG compression [51] can be defined as a function f_Q such that $I' = f_Q(I)$, where I' is the JPEG compressed image, Q is the quantization matrix (8×8) containing the quantization factors $q_i \in \mathbb{N}$ with $i \in \{1, 2, \ldots, 64\}$. As first step, $f_Q(I)$ converts I from the RGB to YCbCr color space, then divides the input image in 8×8 non-overlapping blocks applying also the integer DCT (Discrete Cosine Transform). Finally, each 8×8 block is divided, pixel by pixel, by Q, rounded and then encoded by classic entropy based engine. In this thesis, only the luminance (i.e., Y channel) will be exploited. Let's define also $I'' = f_{Q_2}(f_{Q_1}(I))$ a JPEG double compressed image, with Q_1 and Q_2 denoting the quantization matrices employed for the first and the second compression respectively.

In this thesis, I will refer to QF [1] as the standard quantization matrix associated to a specific quality factor [51] and QF_i to further specify the JPEG compression (e.g. $i = 1, 2, \dots$) in which the matrix was employed. We denote with h_i the empirical distributions built from the i-th DCT coefficients extracted from the 8×8 blocks of I''. Moreover, we define the k quantization factors, in zig-zag order, of Q_1 as $q1_1, q1_2, \dots, q1_k$, and as $q1$ and $q2$ the quantization factors employed in the first and in the second compression respectively.

2.2 State of the art

JPEG is the most studied compression algorithm and the most investigated for digital investigation purposes. When facing a JPEG image, the first problem that could be addressed is the image history reconstruction. The estimation of the matrix employed during the first compression is extremely useful in forensic investigations. Several solutions have been proposed in the years by researchers.

The first approach on this topic was proposed by Fan and De Queiroz in [14] where the authors don't analyze specifically a JPEG image, but a bitmap that has been previously JPEG compressed; they built a method that in the first part determines if the image under analysis has been previously JPEG compressed and then a maximum likelihood based step follows to individuate a compression signature that describes the compression parameters.

Bianchi et al. ([28, 30, 33]) proposed the first robust technique for FQE. The method applies a simple treshhold detector based on Maximum Likelihood to individuate the compression parameters and it is based on a single feature extracted by the analysis on the DCT periodicity between the blocks; moreover the final approach described allows to determinate the grid shift and then to work in the nonaligned scenario.

[1] The notation $QF = x$ referred to Photoshop custom matrices in the next sections is not the standard meaning of quality factor as described before. Photoshop defined 12 different 'quality of compression' referred in this thesis as Photoshop QF.

Li et al.in [52] presented a statistical analysis of JPEG noises (quantization and rounding) during the JPEG compression cycle: the analysis proves that noise distributions are not equal in different steps of compression and that they are dependent on the parameters between two successive compression. The authors built a statistical model based on this analysis demonstrating its goodness also in FQE task.

More researchers in the years analyzed the histograms of the quantized DCT coefficients as features. Galvan et al. [53] analize them and their changes between first and second compression JPEG compression when the second quantization step is lower than the first one; moreover the authors in this work added a proper filter strategy based on residual error removing which improves the results of the estimation. A filter-based selection of DCT histograms strategy was exploited by Dalmia and Okade in [54] and [55] where the authors investigated the impact of blocking artifact through filtering in the nonaligned scenario of double JPEG compression; this investigation with the analysis of residual noise is finally used to compare the filtered histograms with an artificial second quantization matrix in order to estimate the first quantization matrix. In [56], instead the authors analyze the relationship between the first quantization step and the mean square error (MSE) among DCT coefficient histogram bins to extract some candidates and estimate the parameters of first compression. Another work based on the same feature is [57] later extended in [58]: the authors proposed to estimate the quantization factors by finding, among a set of possible candidates, the best matching with the created statistical model in terms of minimal symmetrized Kullback-Leibler (KL) divergence.

The large amount of data and the possibility to analyze it suggested the authors in the year to address the problem with modern Machine Learning (ML) approaches. An initial tentative of estimation by exploiting neural networks was conducted by Lukáš and Fridrich in [38], refined in [59] with error considerations similar to [53]; the authors of [38] transformed the problem of finding the closest histogram into a classification issue, employing a neural network classifier. The introduction of Convolutional Neural Networks (CNNs) represented an additional step in facing this task. CNNs have proven to be amazingly strong at spotting undetectable correlations on data, however, due to the large number of involved parameters, they could suffer from overfitting (i.e., the obtained model is not generalizable enough to represent the phenomenon under analysis); moreover, most deep learning solutions have been designed to work directly on input patches, which limits the usability of the produced model to a very specific scenario. [60, 61, 62] have faced the FQE problem through the use of CNNs, employing different architectures and features. Niu et al. [63] and Tondi et al. [64] faced the problem as a regression one, training their DenseNet architecture based network with a loss function that takes into account both the accuracy and the MSE.

2.3 FQE through a JPEG simulation

In this section, an extensive simulation-based technique is proposed, in this section, focusing on the deduction of the first quantization for a number of Discrete Cosine Transform coefficients by exploiting local image statistics without using any a-priori knowledge. This method is of significant importance for forensic purposes, as it provides a reliable confidence value for the estimation. Experimental results w.r.t. the state-of-the-art demonstrate the effectiveness of the proposed technique both in terms of precision and overall reliability. The method and the extended version have been published ([65, 66]).

Given a double compressed JPEG image I, the main purpose is to estimate the first k quantization factors (zig-zag order) of the first quantization matrix, defined as $\mathbf{q1} = \{q1_1, q1_2, ..., q1_k\}$. Being I doubly compressed means that the only information available is that related to the second compression, which can be related to standard JPEG Quantization tables or custom ones [67, 51]. The second quantization matrix $Q2$ is immediately available by accessing the JPEG file, a procedure performed by using the LibJpeg C library [2] to extract the DCT coefficients of each 8×8 block (D_{ref}). No inverse DCT operation is done at this level, so as not to introduce additional truncation errors. The set of DCT blocks obtained and their respective coefficients (multiplied by $Q2$) are gathered to compute a histogram of the distribution for each of the first k coefficients in zigzag order denoted by: $h_{ref,k}(D_{ref})$ with $k \in \{1, 2, .., 64\}$.

In order to work with the actual contents of image I, it is decompressed with

[2] https://github.com/LuaDist/libjpeg

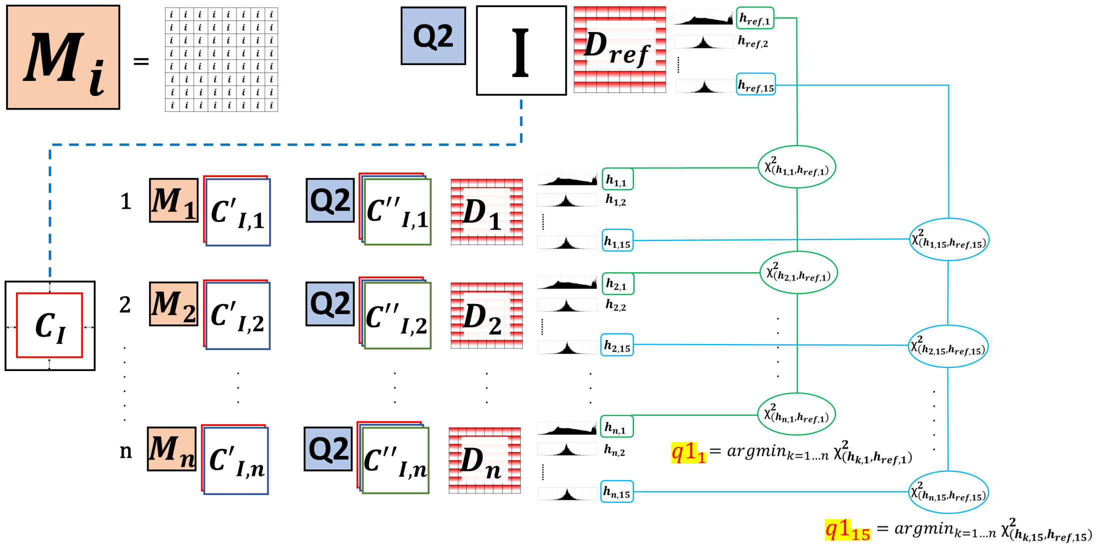
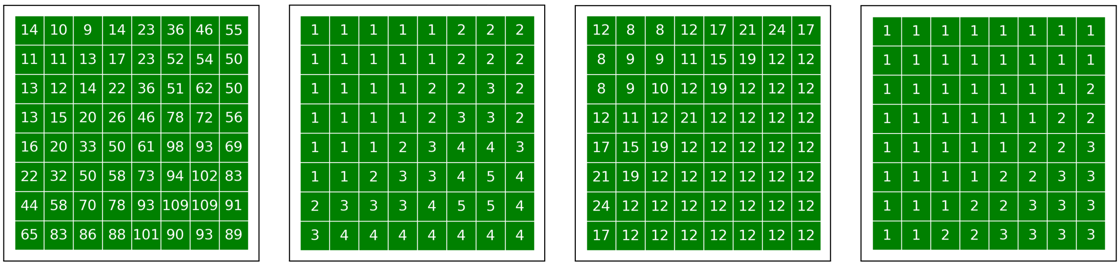

M_i
=
Q2
I
D_{ref}
$h_{ref,1}$
$h_{ref,2}$
$h_{ref,15}$
C_I
1
M_1
$C'_{I,1}$
Q2
$C''_{I,1}$
D_1
$h_{1,1}$
$h_{1,2}$
$h_{1,15}$
2
M_2
$C'_{I,2}$
Q2
$C''_{I,2}$
D_2
$h_{2,1}$
$h_{2,2}$
$h_{2,15}$
n
M_n
$C'_{I,n}$
Q2
$C''_{I,n}$
D_n
$h_{n,1}$
$h_{n,2}$
$h_{n,15}$
$\chi^2_{(h_{1,1},h_{ref,1})}$
$\chi^2_{(h_{2,1},h_{ref,1})}$
$\chi^2_{(h_{n,1},h_{ref,1})}$
$\chi^2_{(h_{1,15},h_{ref,15})}$
$\chi^2_{(h_{2,15},h_{ref,15})}$
$\chi^2_{(h_{n,15},h_{ref,15})}$
$q1_1 = argmin_{k=1...n}\, \chi^2_{(h_{k,1},h_{ref,1})}$
$q1_{15} = argmin_{k=1...n}\, \chi^2_{(h_{k,15},h_{ref,15})}$

Figure 2.3: A schematic representation of the proposed FQE method. The pipeline starts with an Image I to suitably obtain a patch C_I on which the simulation of the first compression with n constant matrices M_i is performed. The results obtained after simulating the first compression, are further compressed with known $Q2$ parameters and on the obtained images the distribution of the DCT coefficients are computed as $h_{n,k}(D_{i,k})$. Each histogram $h_{n,k}(D_{i,k})$ is then compared to a reference one computed on the original I by means of χ^2 distance. The most similar distribution for each element is chosen for the FQE task.

common image readers (python *Pillow* library[3]) and a square patch C_I, with dimensions $d \times d$ with $d \in \{64, 128, 256\}$, is cropped out from the center of I. C_I is then cropped once more, leaving out 4 pixels for each direction, obtaining its new dimension $r \times r$ with $r \in \{56, 120, 248\}$. This is done in order to break the structure of JPEG blocks and remove dangerous correlation noise. C_I is then used as input to simulate the double compression, performed with a number $n > 0$ of constant matrices M_i with $i \in \{1, 2, .., n\}$. The value n (e.g., the worst case) is the largest possible number to be estimated in the range $\{q1_1, q1_2, ..., q1_k\}$: Under the assumption that θ is the lowest quality factor of first compression, n will be the highest value of the first k quantization factors of the quantization table related to the quality factor θ. Having defined the parameter n the double compression simulation of C_I is organized (see Figure 2.3). We define M_i as a quantization matrix 8×8 with each element equal to i and JPEG compress C_I using them so as to generate $C'_{I,i}$

[3] https://pillow.readthedocs.io/en/stable/



14	10	9	14	23	36	46	55
11	11	13	17	23	52	54	50
13	12	14	22	36	51	62	50
13	15	20	26	46	78	72	56
16	20	33	50	61	98	93	69
22	32	50	58	73	94	102	83
44	58	70	78	93	109	109	91
65	83	86	88	101	90	93	89

(a) Standard $QF = 55$

1	1	1	1	1	2	2	2
1	1	1	1	1	2	2	2
1	1	1	1	2	2	3	2
1	1	1	1	2	3	3	2
1	1	1	2	3	4	4	3
1	1	2	3	3	4	5	4
2	3	3	3	4	5	5	4
3	4	4	4	4	4	4	4

(b) Standard $QF = 98$

12	8	8	12	17	21	24	17
8	9	9	11	15	19	12	12
8	9	10	12	19	12	12	12
12	11	12	21	12	12	12	12
17	15	19	12	12	12	12	12
21	19	12	12	12	12	12	12
24	12	12	12	12	12	12	12
17	12	12	12	12	12	12	12

(c) Photoshop $QF = 5$

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2
1	1	1	1	1	1	2	2
1	1	1	1	1	2	2	3
1	1	1	1	2	2	3	3
1	1	1	2	2	3	3	3
1	1	2	2	3	3	3	3

(d) Photoshop $QF = 12$

Figure 2.4: Example of the standard quantization matrices of $QF = 55$ (a) and $QF = 98$ (b), and custom quantization matrix extracted from Photoshop of $QF = 5$ (c) and $QF = 12$ (d); Photoshop defines its custom table with quality factor in range [1,12].

with $i \in \{1, .., n\}$. A second compression can now be simulated by using the known $Q2$ on each of the n $C'_{I,i}$ thus generating $C''_{I,i}$ new compressed images. Each $C''_{I,i}$ represent a simulation of double compression with known first and second quantization parameters. At this point, for every $C''_{I,i}$, it is possible to extract the DCT coefficients $D_{i,k}$ and in the same way as described before, a distribution for $D_{i,k}$ is then computed and defined through the generation of $h_{n,k}(D_{i,k})$.

$h_{n,k}(D_{i,k})$ is a set of n distributions representing all the n simulations of double compression. These simulated distributions $h_{n,k}(D_{i,k})$ are then compared one by one with the real one $h_{ref,k}(D_{ref})$ through χ^2 distance:

$$\chi^2(x, y) = \sum_{i=1}^{m} (x_i - y_i)^2/(x_i + y_i) \tag{2.1}$$

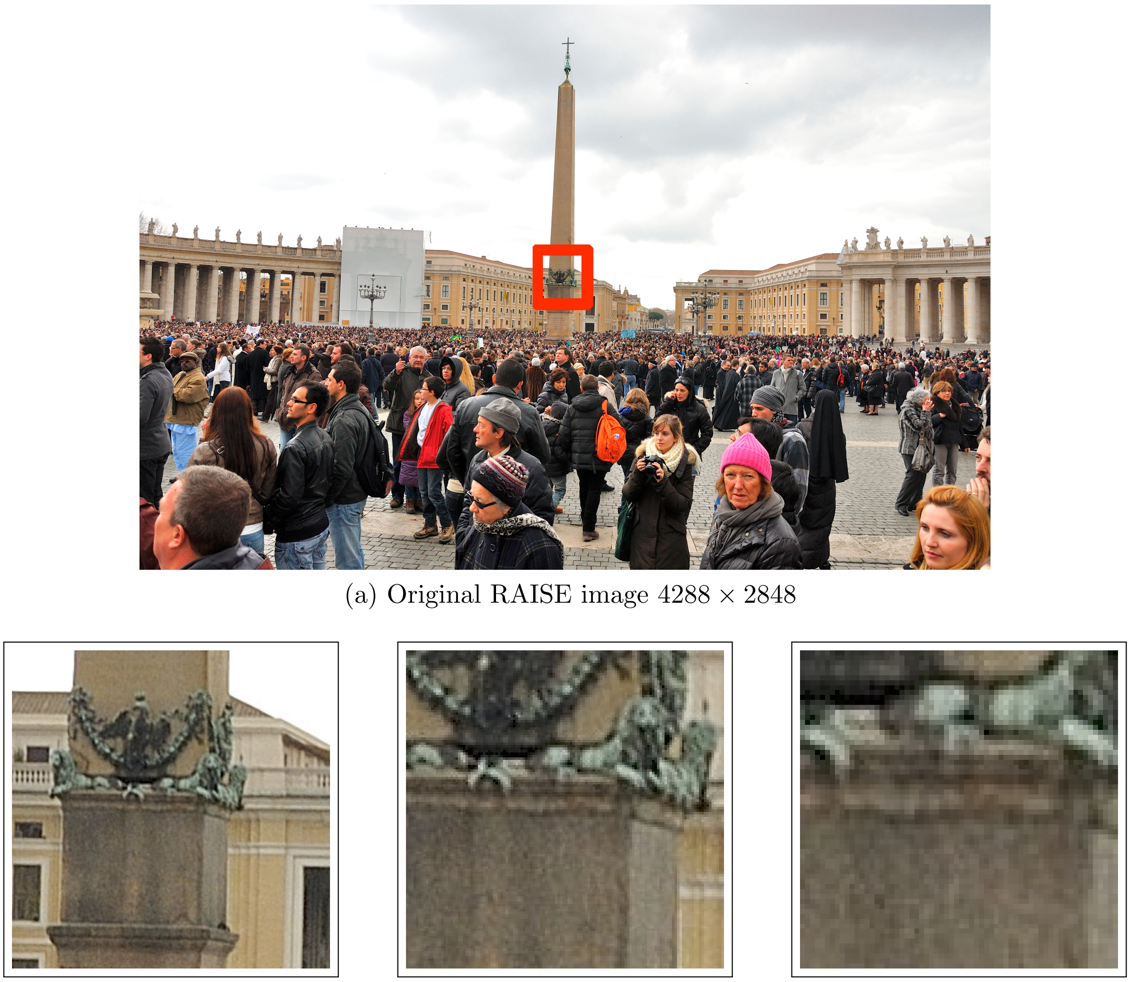
where x and y represent the distribution to compare.

Finally the actual estimation of **q1**, which is the main goal of the described technique, can be done as follows:

$$q1_k = argmin_{k=1,..,n}\chi^2(h_{n,k}(D_{i,k}), h_{ref,k}(D_{ref})) \tag{2.2}$$

For sake of clarity, the pseudo-code of the process is reported in Algorithm 1.

The effectiveness of the proposed approach, was demonstrated through experiments performed on the RAISE dataset [68], composed by 8156 high resolution images in TIFF format (uncompressed). An example of a RAISE image is shown in Figure 2.5a. For testing purposes, patches of different dimensions were extracted



(a) Original RAISE image 4288 × 2848

(b) Crop 256 × 256

(c) Crop 128 × 128

(d) Crop 64 × 64

Figure 2.5: Example of a RAISE image and the cropping operation on it. The center usually contains the most informative part of the image, which is desirable for FQE methods.

from the original ones as shown in Figures 2.5b, 2.5c, 2.5d. The patches were obtained by extracting an appropriate region from the central part of the original images as the main focus of the observer is often directed to that point which contains most of the information. A double-compressed JPEG image dataset was then created from the TIFF cut images while the test was generated as a combination of three parameters: crop size $d \in \{64, 128, 256\}$, first and second compression quality factor $QF_1 \in \{50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$ $QF_2 \in \{80, 90\}$. Although RAISE images have been cropped, the number of blocks b analyzed for each image remains fairly representative: $b \in \{64, 256, 1024\}$ blocks for $d \in \{64, 128, 256\}$

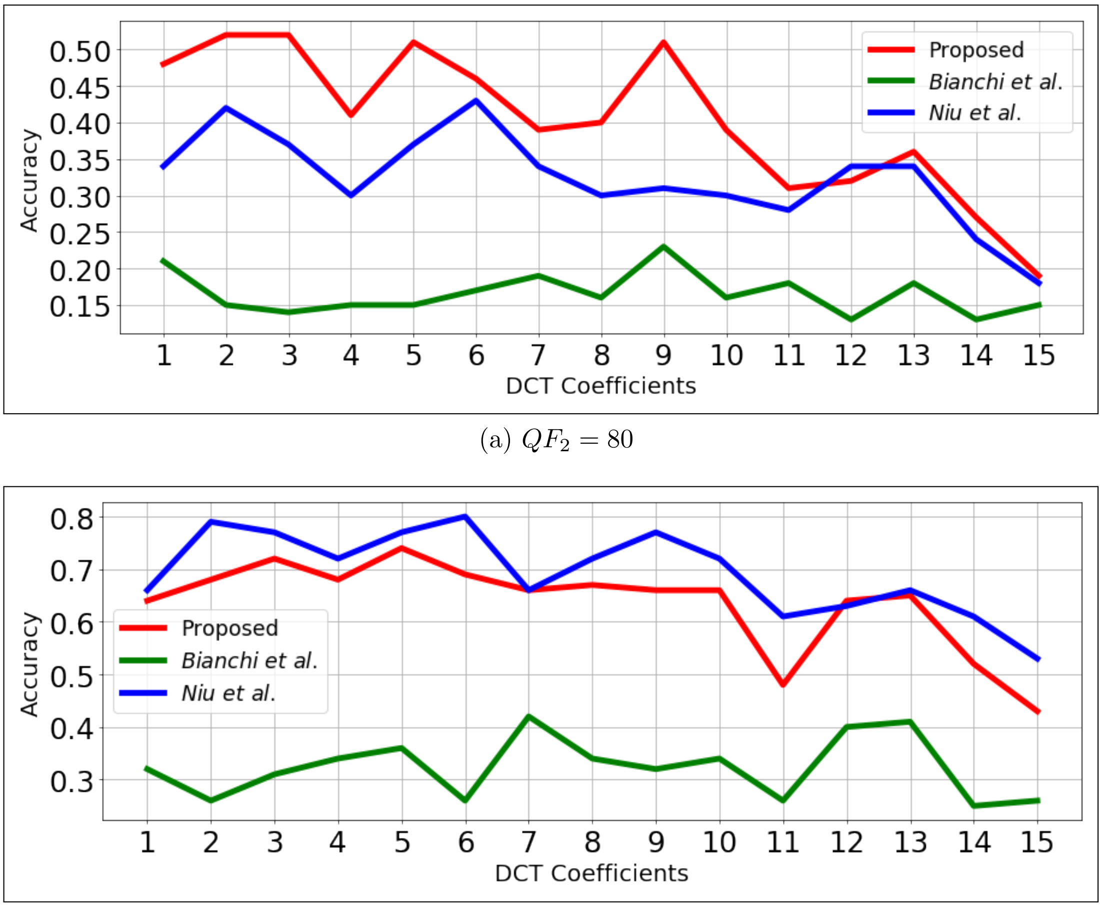
Algorithm 1 FQE method through JPEG simulation.

```
Input: $m$-compressed image $I$
Output: $q_{m-1} = \{q_1, q_2, ......, q_k\}$
Initialization : $k$, $n$
for $j = 1$ to $k$ do
    $h_{ref,j}(D_{ref})$ : distribution of $j$-th DCT coefficient of $I$
end for
$q_m$ : known compression matrix of $I$
$C_I$ : $r \times r$ misaligned crop from $I$
for $i = 1$ to $n$ do
    $C'_{I,i}$ : compression of $C_I$ with constant matrix $M_i$
    $C''_{I,i}$ : compression of $C'_{I,i}$ with $q_m$
    $D_i$ : DCT coefficients of $C''_{I,i}$
    for $j = 1$ to $k$ do
        $h_{i,j}(D_i)$ : distribution of $j$-th coefficient of $D_i$
    end for
end for
for $j = 1$ to $k$ do
    $q_j$ : lower $\chi^2$ distance between $h_{ref,j}(D_{ref})$ and $h_{i,j}(D_i)$ with $i \in \{1, 2, ..., n\}$
end for
return $q_{m-1} = \{q_1, q_2, ......, q_k\}$
```

respectively.

All simulations were employed working on standard quantization tables considering only the luminance factor. To evaluate the generality of the proposed strategy, Photoshop custom tables (Adobe Photoshop CC version 20.0.4) were also employed. To be aware about the inherent consistency and availability of a certain amount of data, the range of the quantization factor in this case is $QF_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$. Figure 2.4 shows the employed tables by considering the limit cases (low and high) in both cases (standard, custom). Unlike most state-of-the-art methods, our statistical analysis is completely independent of the specific quantization table because it considers the statistics of each coefficient.

Let's define the actual dataset as $Dataset_{d,QF_1,QF_2}$ which represents RAISE center-cropped images of size $d \times d$, compressed the first time with quality factor QF_1, and the second time with quality factor QF_2. In order to generate each element of $Dataset_{d,QF_1,QF_2}$, all JPEG compressions were carried out directly on TIFF images using the python image library *Pillow*.



(a) $QF_2 = 80$

(b) $QF_2 = 90$

Figure 2.6: Average accuracy of the estimation for each DCT coefficient employing standard tables. (a) shows results with $QF_2 = 80$. (b) shows results with $QF_2 = 90$. The comparison was done with Bianchi et al. [33] and Niu et al. [63].

2.3.1 Experimental results

The approach described before is the result of a set of experiments conducted on images in $Dataset_{d,QF_1,QF_2}$ obtained from the RAISE dataset. In particular the χ^2 distance was chosen as the best, in terms of results produced, among a set of 12 different distances tested [4]. Moreover, the $L1$ and the $L2$ normalizations did not change the results obtained on tested data, thus data normalization was not included into the proposed approach.

[4] 1. Chi squared (χ^2), 2. Intersection, 3. Manhattan, 4. Braycurtis, 5. Cosine, 6. Correlation, 7. Euclidean, 8. Minkowski, 9. Chebyshev, 10. Canberra, 11. Hamming, 12. Jaccard

QF_1	$QF_2 = 80$					$QF_2 = 90$				
	Proposed	*[33]*	*[53]*	*[54]*	*[63]*	*Proposed*	*[33]*	*[53]*	*[54]*	*[63]*
55	**0.57**	0.42	0.43	0.5	0.28 9	-	-	-	-	-
60	0.54	0.31	0.42	0.49	**0.6**	**0.76**	0.66	0.63	0.68	0.73
65	**0.58**	0.21	0.47	0.54	0.37	**0.77**	0.66	0.65	0.69	0.6
70	**0.66**	0.2	0.56	0.59	0.52	**0.78**	0.51	0.64	0.72	0.74
75	0.52	0.09	**0.61**	0.49	0.19	0.82	0.74	0.47	0.69	**0.85**
80	**0.04**	0	0	0	0.01	0.82	0.31	0.68	0.65	**0.88**
85	**0.33**	0.16	0	0	0.05	0.75	0.14	0.78	0.47	**0.85**
90	0.22	0.06	0	0	**0.5**	**0.23**	0	0	0	0.03
95	0.17	0.02	0.01	0.02	**0.43**	0.38	0.05	0.01	0	**0.75**
98	-	-	-	-	-	0.05	0.02	0.02	-	**0.75**
MEAN	**0.40**	0.16	0.28	0.29	0.33	0.63	0.32	0.46	0.49	**0.69**

Table 2.1: Accuracy obtained by proposed approach compared to Bianchi et al. [33], Galvan et al.[53], Dalmia et al.[54] and Niu et al. [63] with different combinations of QF_1/QF_2 and employing the corresponding standard tables.

The best pipeline among all tests, was employed for comparison against the state-of-the-art and is described in Figure 2.3. Four approaches were considered for comparisons: Bianchi et al. [33], that is a milestone among analytical methods; Galvan et al. [53] and Dalmia et al. [54] which achieve state of the art results when $QF_1 < QF_2$ and Niu et al. [63], which represents the state-of-the-art employing CNNs and achieving the best results. It is worth noting that Niu et al. [63] uses different trained neural models for each QF_2 (80 and 90), while the proposed solution works for any QF_2 with the same technique.

As regards implementations used for testing the above mentioned techniques: the publicly available[5] Matlab implementation was employed to replicate Bianchi et al. [33]; code from the ICVGIP-2016.RAR archive available on Dr. Manish Okade's website[6] was employed for testing Dalmia et al. [54]; models and implementation available on Github[7] were employed for tests on Niu at al. [63] and finally an implementation from scratch was employed for results for Galvan et al. [53]

All tests were performed with $k = 15$ for comparison purpose but the method could be easy generalized for any $0 \leq k \leq 64$, even if the theoretical limit would become increasingly lower. However, $k = 15$ is tipically employed in literature,

[5] http://lesc.det.unifi.it/en/node/187

[6] https://sites.google.com/site/manishokade/publications

[7] https://github.com/andreacos/CnnJpegPrimaryQuantizationEstimation

QF_1	$QF_2 = 80$				$QF_2 = 90$			
	Proposed	*[33]*	*[53]*	*[63]*	*Proposed*	*[33]*	*[53]*	*[63]*
5	**0.66**	0.39	0.53	0.08	**0.78**	0.64	0.66	0.07
6	0.45	0.11	**0.46**	0.03	**0.82**	0.51	0.68	0.08
7	**0.62**	0.23	0.54	0.08	**0.78**	0.5	0.64	0.08
8	**0.19**	0.06	0.04	0.02	**0.79**	0.26	0.7	0.1
9	**0.3**	0.19	0	0.09	**0.49**	0.03	**0.49**	0.02
10	0.18	0.01	0	**0.41**	**0.43**	0.18	0	0.28
11	0.19	0.04	0.04	**0.26**	0.39	0.05	0.02	**0.7**
12	0.21	0.05	0.05	**0.22**	0.38	0.05	0.02	**0.75**
MEAN	**0.35**	0.14	0.2	0.15	**0.61**	0.28	0.4	0.26

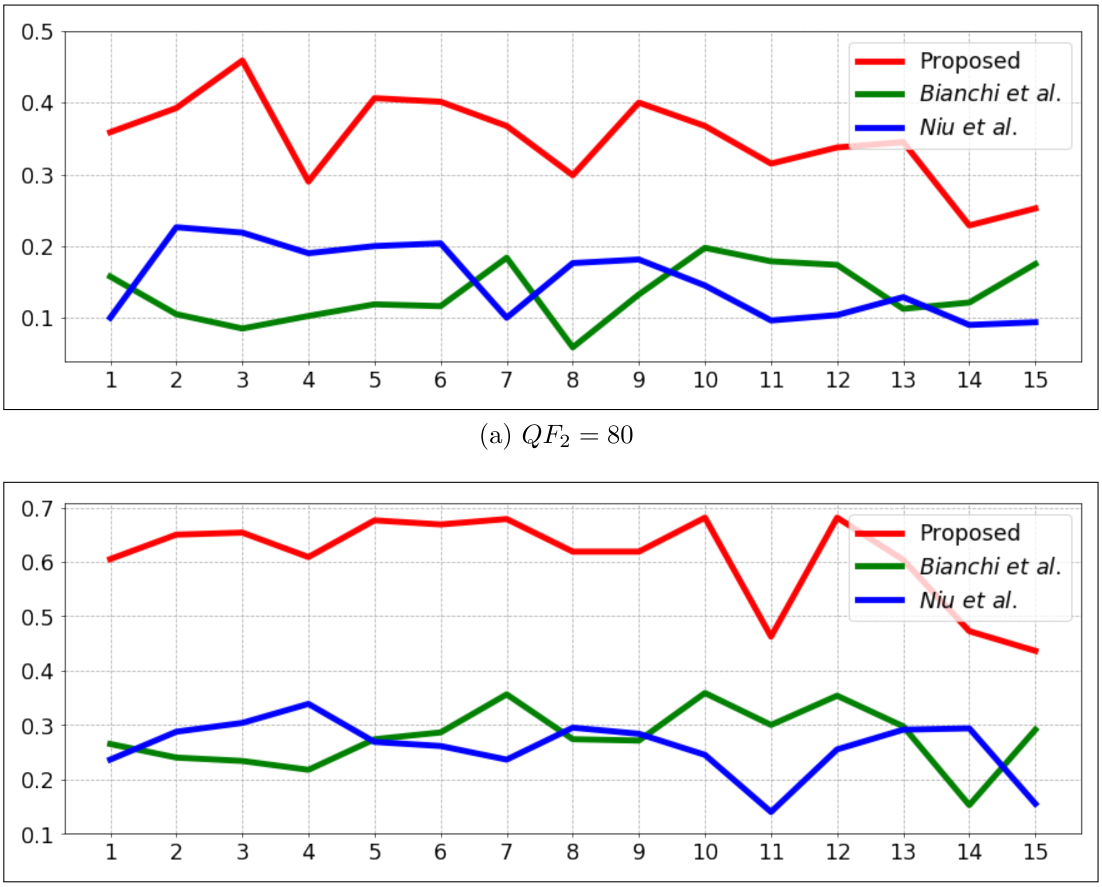
Table 2.2: Accuracy obtained by proposed approach compared to Bianchi et al. [33], Galvan et al.[53] and Niu et al. [63] employing the custom tables for first compression. The column QF_1 refers to Photoshop's custom tables. In this test Dalmia et al. [54] was not considered given their implementation explicitly asks for standard tables as input.

which is why we used the same for comparison purpose.

A first comparison was performed taking into account standard JPEG quantization tables for primary quantization: accuracy results are reported in Table 2.1. Specifically, the standard tables related to the following quality factors were employed: $QF_1 \in \{55, 60, 65, \ldots, 95\}$ with $QF_2 = 80$ and $QF_1 \in \{60, 65, \ldots, 90\}$ with $QF_2 = 90$. Thus, the two employed datasets were $Dataset_{64,\{55,60,65,70,75,80,85,90,95\},80}$ and $Dataset_{64,\{60,65,70,75,80,85,90,95,95\},90}$ respectively with a total of 73413 JPEG double compressed images for each set. The proposed approach was employed with $n = 22$ for tests on the first dataset and $n = 19$ on the second one.

Table 2.1 shows that the proposed approach outperforms all other methods while there is an alternation in best results with Niu et al. [63]. Specifically the proposed approach shows better results for lower QF_1 and $QF_2 = 80$.

The gap in results obtained compared to the state-of-the-art changes greatly when considering custom tables, which are the most common in a real forensic investigation scenario. Photoshop was employed in order to generate double compressed images as described before, creating another dataset of 65256 JPEG double-compressed images. Table 2.2 shows the results achieved. In this case, the proposed method not only outperforms the state-of-the-art, but also confirms the same robust behaviour obtained with tests on standard tables as shown in Figures 2.6-2.7: both



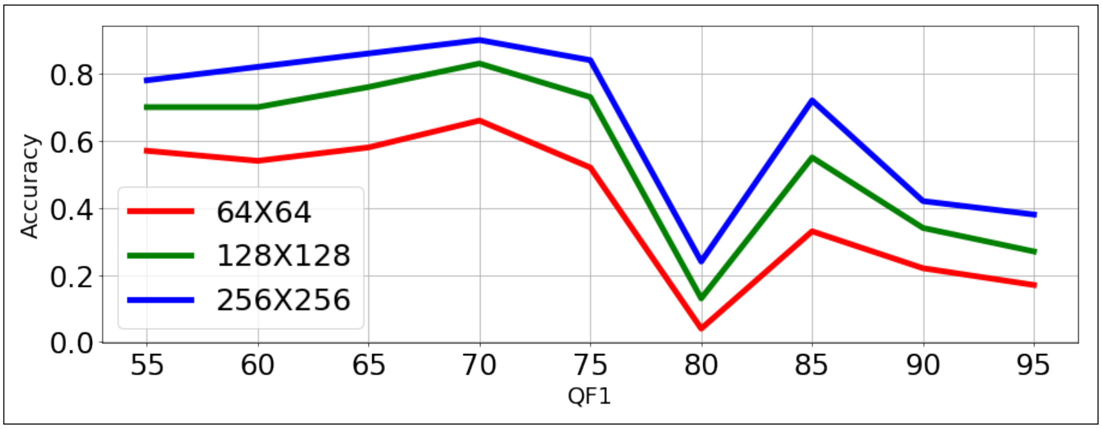
(a) $QF_2 = 80$

(b) $QF_2 = 90$

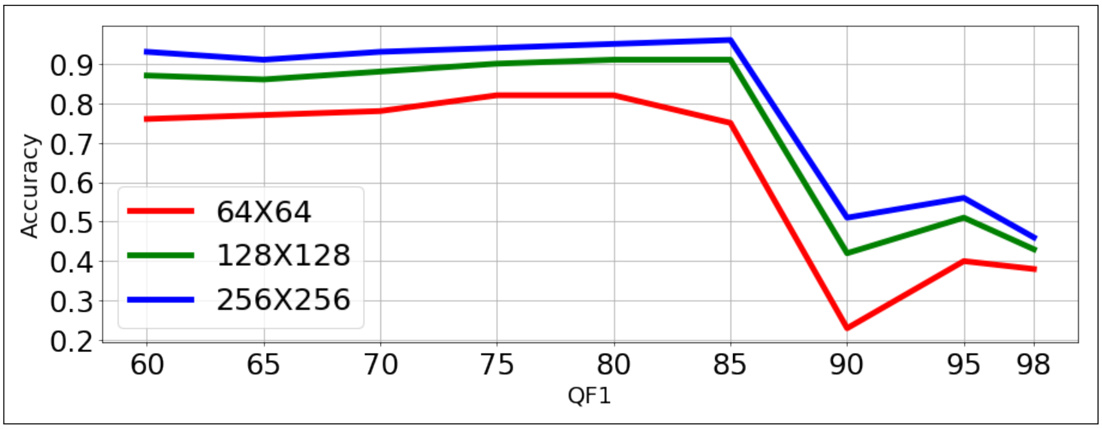
Figure 2.7: Average accuracy of the estimation for each DCT coefficient employing custom tables. (a) shows results obtained with $QF_2 = 80$. (b) shows results obtained with $QF_2 = 90$. The comparison was done with Bianchi et al. [33] and Niu et al. [63].

trends are very similar for the proposed approach while other methods register a decrease in their accuracy levels (axis y).

The patch size is indeed a relevant parameter. Figure 2.8 shows a trend already demonstrated in the state-of-the-art in which, given the same QF conditions, the accuracy obtained by any method depends almost proportionally on the patch size of the input image, or rather, on the overall amount of information contained in it. Giving the importance of the patch size, in terms of results, a comparison with Thai et al. [58] was carried out. Thai et al. proposed a method with results demonstrated on bigger images and patch sizes. In order to have a fair comparison, the same images from the BOSSBase [69] dataset were considered, employing a patch size 512 × 512



(a) $QF_2 = 80$



(b) $QF_2 = 90$

Figure 2.8: Accuracy obtained by the proposed method at different patch size with $QF_2 = 80$ (a) and $QF_2 = 90$ (b). Standard tables were employed.

on a total of 10000 images. Thai et al. [58] estimate only the first 10 quantization factors (zig-zag order) at specific conditions (no estimation is done when multiples are involved). Thus, we employed the following parameters for testing: $QF_1 \in \{60, 65, 70, 75, 80, 85\}, QF_2 = 90$, which are the only QF combinations were Thai et al. estimate all the first 10 factors, $n = 19$ and of course $K = 10$. The results obtained from the proposed approach, in terms of the average accuracy across all factors and all QF_1, is 99.7%, which is the same as Thai et al. as reported in [70].

In order to prove the robustness of the proposed solution w.r.t. image contents and acquisition conditions (e.g., different devices), further tests have been performed. Specifically, three datasets have been considered: Dresden [71], UCID [72] and BOSSBase [69]. Results reported in Tables 2.3,2.4 and 2.5, confirm the

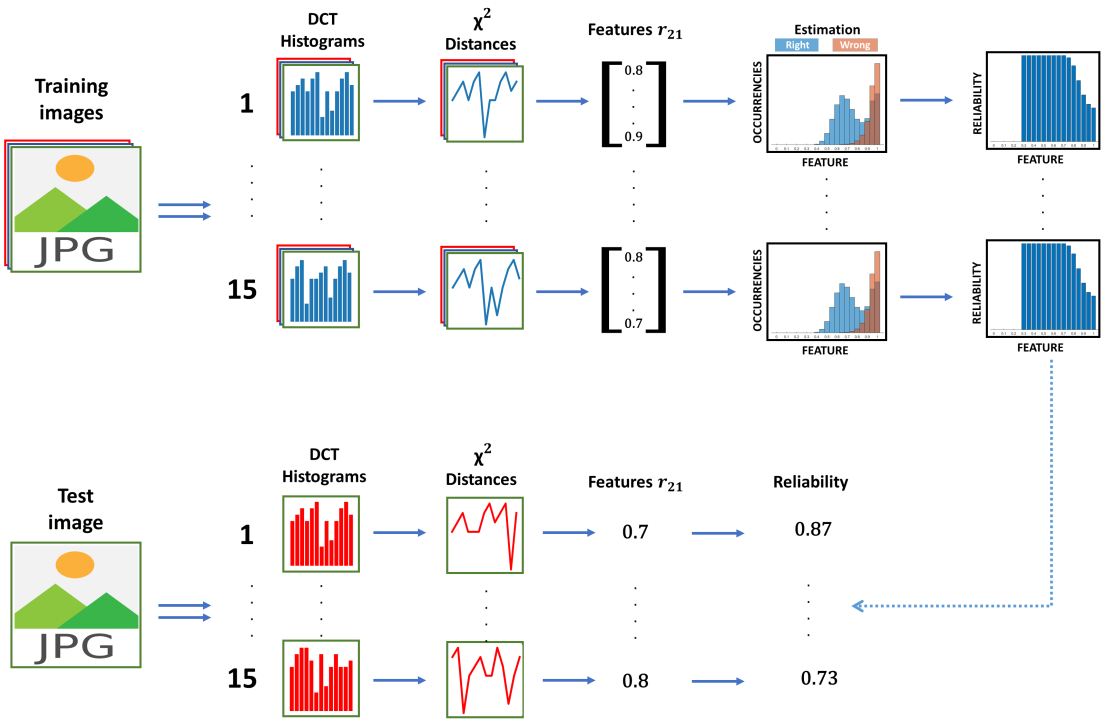


Figure 2.9: Overall scheme of the reliability index computation. As first, training images are collected and taken as input from the proposed FQE algorithm. Then, r_{21} feature is computed from the χ^2 difference curves. Distributions of right and wrong estimations with respect to the feature r_{21} can be derived considering also ground truth labels. Finally, the reliability index curves are simply obtained by the ratio of the correct to the overall estimations for a fixed range of r_{21}. Given a test image, the reliability of the performed predictions can be simply obtained from the the comparisons of features r_{21} values and the reliability index curves generated from training images.

Images	**Size**	**Patch size**	**%**		**QF1**								
				QF2	60	65	70	75	80	85	90	95	100
1488	160X120	64X64	21,3%	90	0,96	0,96	0,94	0,93	0,91	0,75	0,21	0,33	0,30
		64X64	21,3%	80	0,78	0,73	0,76	0,60	0,05	0,29	0,19	0,13	0,14
		160X120	100%	90	0,99	0,99	0,99	0,99	0,99	0,97	0,44	0,32	0,32
		160X120	100%	80	0,91	0,87	0,94	0,89	0,84	0,21	0,21	0,15	0,16

Table 2.3: Accuracy obtained by the proposed approach on Dresden [71] dataset with different patch size and QF_1/QF_2. The % value represents the percentage of crop size compared to the original size.

Images	**Size**	**Patch size**	**%**		**QF1**								
				QF2	60	65	70	75	80	85	90	95	100
1334	512X384	64X64	2,1%	90	0,92	0,93	0,92	0,92	0,90	0,77	0,21	0,35	0,32
		64X64	2,1%	80	0,74	0,72	0,76	0,61	0,04	0,32	0,21	0,14	0,14
		128X128	8,3%	90	0,98	0,98	0,98	0,98	0,98	0,94	0,42	0,51	0,41
		128X128	8,3%	80	0,91	0,92	0,93	0,87	0,14	0,58	0,36	0,24	0,23
		256X256	33,3%	90	0,99	0,99	0,99	0,99	0,99	0,99	0,51	0,58	0,44
		256X256	33,3%	80	0,97	0,97	0,98	0,97	0,28	0,78	0,42	0,36	0,33
		512X384	100%	90	0,99	0,99	0,99	0,99	0,99	0,99	0,59	0,54	0,42
		512X384	100%	80	0,98	0,98	0,99	0,98	0,40	0,83	0,38	0,40	0,34

Table 2.4: Accuracy obtained by the proposed approach on UCID [72] dataset with different patch size and QF_1/QF_2. The % value represents the percentage of crop size compared to the original size.

Images	**Size**	**Patch size**	**%**		**QF1**								
				QF2	60	65	70	75	80	85	90	95	100
10000	512X512	512X512	100%	90	0,99	0,99	0,99	0,99	0,99	0,99	0,66	0,47	0,37
		512X512	100%	80	0,96	0,96	0,99	0,98	0,41	0,76	0,36	0,41	0,33

Table 2.5: Accuracy obtained by the proposed approach on BOSSBase [69] dataset with different patch size and QF_1/QF_2. The % value represents the percentage of crop size compared to the original size.

Low/Low	**Low/Mid**	**Low/High**	**Mid/Low**	**Mid/Mid**	**Mid/High**	**High/Low**	**High/Mid**	**High/High**
0,73	0,8378	0,93	0,5933	0,81	0,9322	0,2811	0,38	0,8189

Table 2.6: Accuracy of proposed approach using RAISE full-size images compressed with custom table from Park et al. [73]

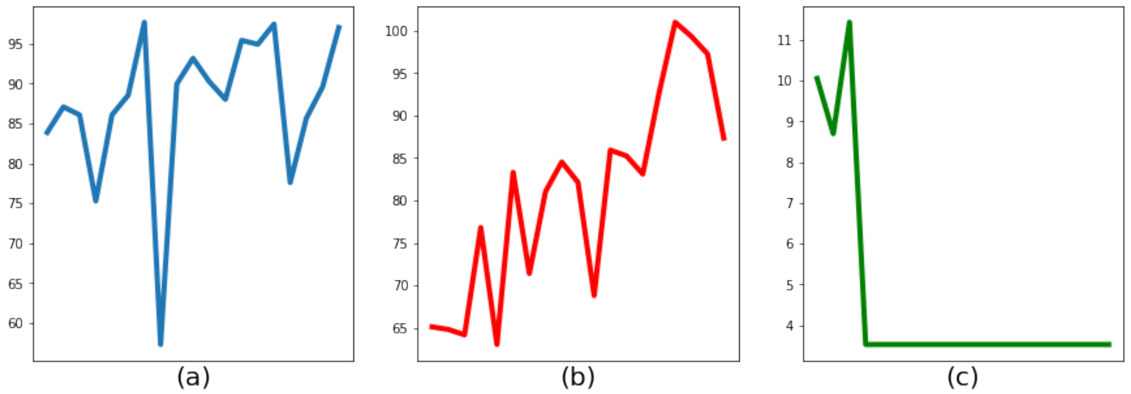


Figure 2.10: χ^2 distance curves examples: (a) curve with a strong minimum; (b) curve containing several local minima with similar values; (c) nearly flat curve.

validity of the proposed method. The impact of the resolution/crop pair is evident observing the results of a single dataset (Table 2.4), where for each increase in crop size (incrementally) corresponds an improvement of accuracy. At the same time, considering the crop from different datasets (64×64 in Tables 2.1,2.3,2.4) the best results are achieved in the crop taken from the lowest resolution dataset. A $d \times d$ crop extracted from a high resolution image contains less information than the one extracted from a smaller image, providing a flatter histogram that is difficult to discriminate.

A final test regarding double compressed images has been performed in a much more challenging scenario: a dataset of 500 full-size RAISE images was employed for first and second compression by using custom tables collected by Park et al. [73]. The authors of [73] collected a dataset of JPEG quantization matrices employed in real scenarios. The collection consists of 1170 different matrices: 1070 custom and 100 standard JPEG quantization tables. For this test, the parameter of the proposed approach was $n = 136$ which is the maximum value of the first 15 coefficients among the 1070 custom quantization tables in this context. Results obtained, in terms of accuracy, are reported in Table 2.6 and definitively demonstrate the robustness of the technique even in a *wild* scenario of non-standard tables.

FQE algorithms estimate the coefficients of the first quantization matrix without providing any information about the related reliabilities. In the forensic domain, it could be fundamental to measure such degree of uncertainty whenever it is available. One of the contributions of this thesis is a statistically robust analysis conducted

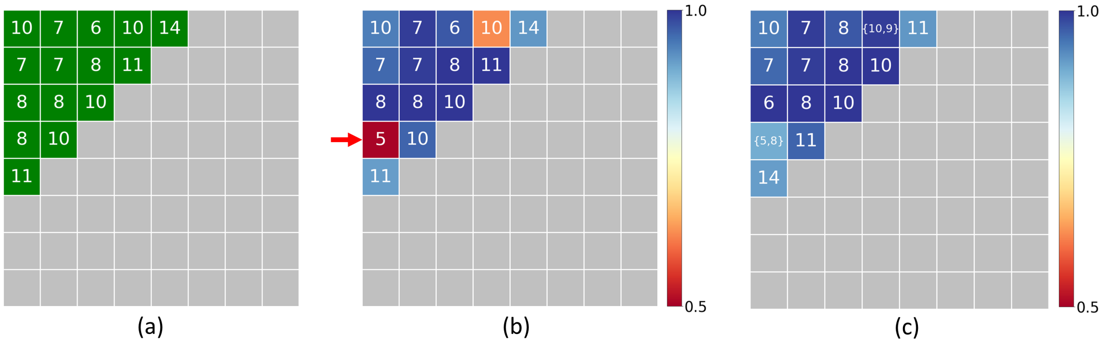


Figure 2.11: An example of ground truth values (a) and the related estimations (b) obtained considering the proposed solution. In (c) is also reported a possible output where the 7^{th} and 10^{th} coefficient (zig-zag order) are pointed-out considering two values augmenting the overall reliability of the estimation. Background colors in (b) and (c) represent the reliability associated to the predictions. The red arrow indicates the wrong estimation.

on the amount of information contained in the images, that allowed to generate a reliability index. To this aim the RAISE dataset [68], including 8156 high resolution 4356×3511 (W, H) mean size images together with the proposed FQE solution are exploited. Specifically, for each source image we generated 73413 (N) crop-center patches of size 64×64 (W', H') compressed twice with different quantization matrices. Moreover, from the analysis of the χ^2 distance curves generated by the double compression simulations, three classes have been clearly identified (see Figure 2.10): (i) curves with a strong minimum; (ii) curves containing several local minima with similar values; (iii) nearly flat curves.

To detect the first type of curves, the only ones able to provide a reliable estimation, a feature based on the ratio between the two lowest values of the aforementioned curve has been designed (r_{21}). A Curve with a strong minimum have r_{21} value close to zero. On the contrary, curves containing local minima with similar values have r_{21} values close to 1. To detect nearly flat curves, in our tests the differences with respect to the median value have been computed and if more than 1/3 of them are zero, the curve is considered nearly flat and the r_{21} is set to 1 (low reliability).

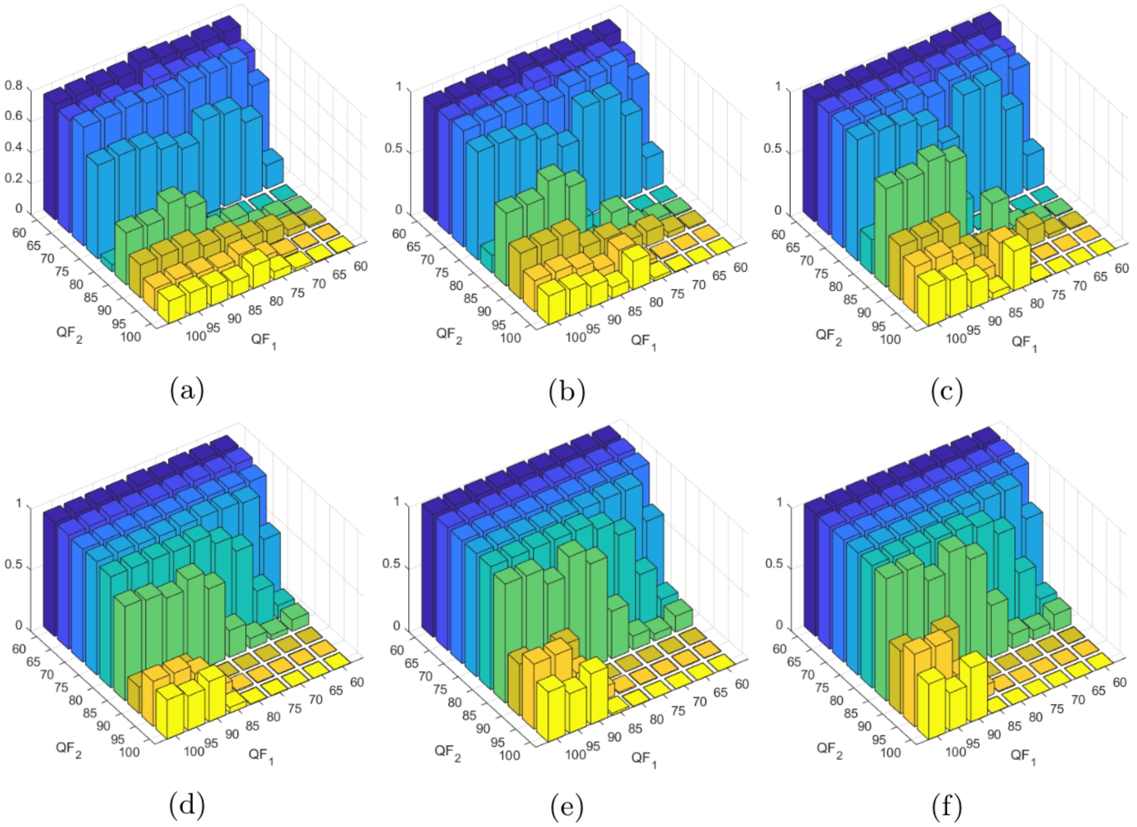
The overall pipeline designed to provide a reliability index to the performed estimations is described in Figure 2.9. A set of training images, collected considering the testing scenario (e.g., image resolution, crop size, employed $Q2$), is considered as input of the proposed algorithm. Starting from the χ^2 difference curves, r_{21} feature is computed (one value per curve). These data, combined with ground truth labels, are then exploited to generate the distributions of right and wrong estimations with respect to the feature r_{21}. Finally, the reliability index curves (one per DCT coefficient) are simply obtained by the ratio of the correct to the overall estimations for a fixed range of r_{21}. Considering then a test image, the reliability of the algorithm predictions can be simply obtained computing the features r_{21} and comparing these values with the reliability index curves previously generated from training images. In Figure 2.11 an example of reliability indexes computed for the first 15 DCT coefficients on a double compressed image with $QF_1 = 70$ and $QF_2 = 90$ is reported. The predicted quantization factors are {10,7,7,8,7,6,10,8,8,5,11,10,10,11,14} whereas ground truth labels are {10,7,7,8,7,6,10,8,8,8,11,10,10,11,14}. It is worth noting that, although the proposed approach provide a wrong estimation of the quantization factor related to the 10^{th} DCT coefficient, the associated reliability index is pretty low (i.e., 0.5). This value indicates that in the training dataset, only half of the images with a similar r_{21} have been correctly estimated.

If the reliability index associated to the performed estimation does not satisfy the requirement of a specific forensics scenario, that estimation can be discarded or further analysis can be done. Specifically, the statistics about the reliability of the estimation computed from training images, can be performed also considering not only the minimum of the χ^2 distance curve, but taking into account the two lowest values of the aforementioned curve. Considering previously described example, and fixed a reliability of 0.9 (i.e., 90% of quantization factors correctly estimated in the same scenario), the proposed solution provides the following estimation: {10,7,7,8,7,6,[10,9],8,8,[5,8],11,10,10,11,14}, where suggesting the pairs [10, 9] and [5, 8] we obtain the increase of reliability to reach the fixed threshold.

Reliability of the estimation could support further analysis in the field refining existing methods especially when such information is used directly for image ballistics purposes ([74], [75]).

2.3.2 Experiments with Multiple Compressions

The hypothesis that only one compression was performed before the last one could be a strong limit. Thus, a method able to extract information about previous quantization matrices, in a multiple compression scenario, may be a considerable contribution. For this reason, the proposed approach was tested in a triple JPEG



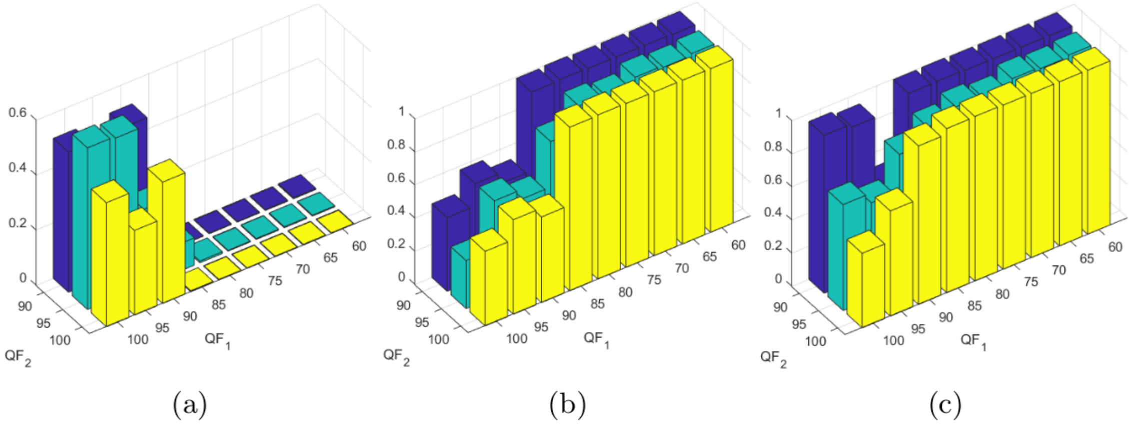
(a) (b) (c)

(d) (e) (f)

Figure 2.12: Overall accuracy of the proposed method on JPEG triple compressed images when trying to estimate the Q_{m-1} quantization factors. First row identifies patch size 64×64, 128×128, 256×256 and $QF_3 = 80$ respectively [(a),(b),(c)], while second row is related to the same patch sizes and $QF_3 = 90$ [(d),(e),(f)

compression scenario, where the new goal was the estimation of the quantization factors related to the second compression matrix. Figure 2.12 shows the accuracy obtained employing different crop sizes (64×64, 128×128, 256×256) on all the combinations $QF_1/QF_2/QF_3$ with $QF_1/QF_2 \in \{60, 65, 70, 75, 80, 85, 90, 95, 100\}$ and $QF_3 \in \{80, 90\}$ with the method that predicts the firsts 15 coefficients of QF_2.

As shown in Figure 2.12, the method in general achieves satisfactory results. Some limits are visible when the first compression is strong (low QF) and the second one has been performed with an high quality factor $QF_2 \in \{90, 95, 100\}$. By analyzing the results in these particular cases, it is worth noting that the method estimates QF_{m-2} instead of QF_{m-1} (where m represents the compression step). Figure 2.13 shows the accuracies obtained in these last cases ($QF_2 \in \{90, 95, 100\}$) considering as correct estimations the quantization factors related to Q_{m-1} (a), Q_{m-2} (b)

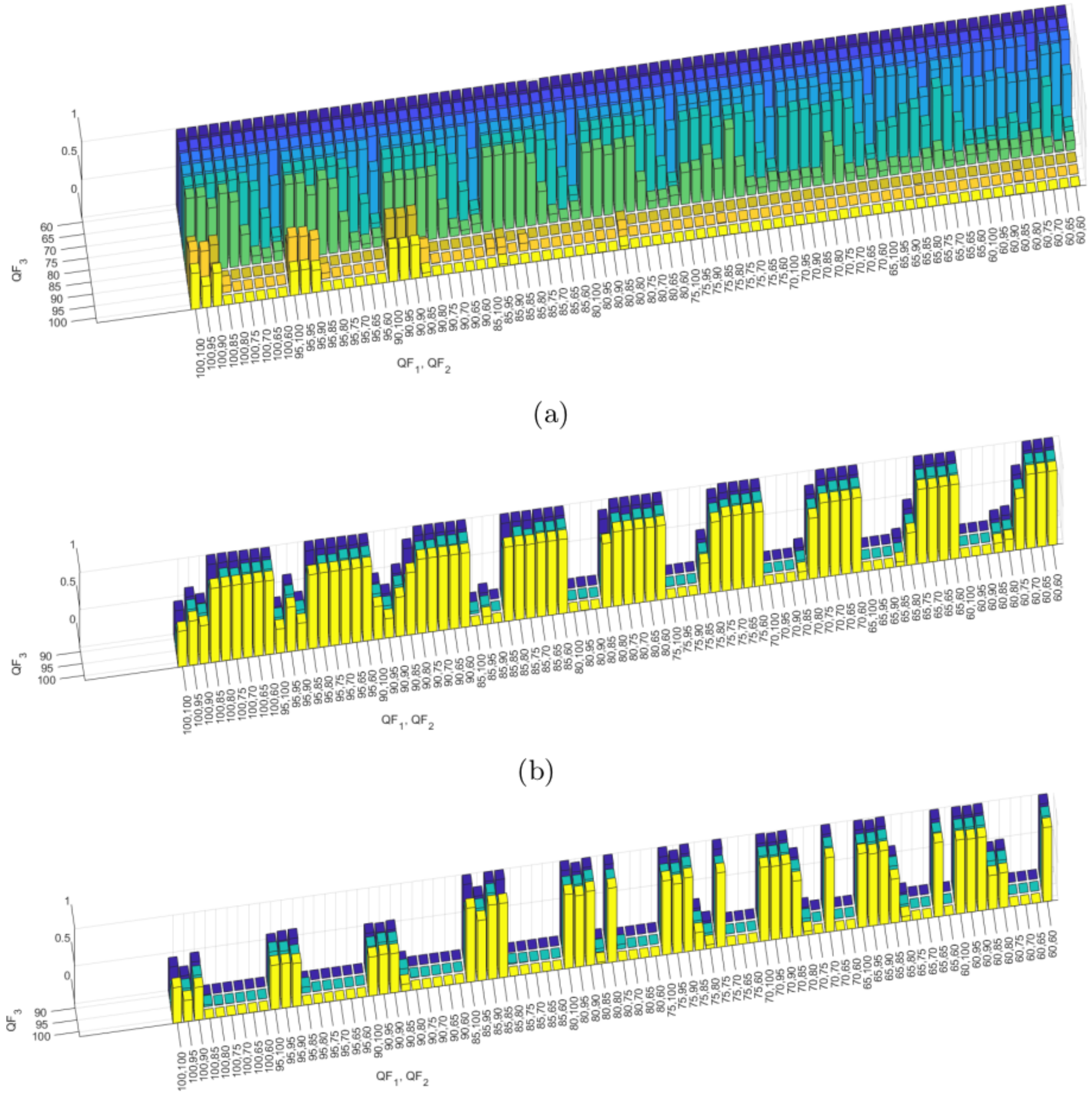


(a) (b) (c)

Figure 2.13: Overall accuracy of the proposed method on JPEG triple compressed images with high QF_2 (90,95,98), patch size 256×256 and $QF_3 = 90$, considering as ground truth (i.e., correct estimations) the quantization factors related to QF_2 (a), QF_1 (b) and both (c).

and both (c). Results shown in (c) demonstrate how the method is able to return information about quantization factors (not only $m - 1$) even in this challenging scenario. Starting from this phenomenon, in order to discriminate a predicted factor q_k between Q_{m-2} and Q_{m-1}, a simple test has been carried out on 100 triple compressed images with $QF_1 = 65$, $QF_2 = 95$ and $QF_3 = 90$. Starting from the cropped image C_I (see Section 2.3), we simulated, similarly to the case of double compressions in the proposed approach, all the possible triple compressions taking into account only two hypothesis (i.e., q_k belongs to Q_2 or Q_1) and considering a constant matrix built from q_k as Q_1 or Q_2 respectively. Thus, the obtained simulated distributions are compared with the real one through χ^2 distance (2.4). In this scenario, the proposed solution correctly estimated Q_1 quantization factors with an accuracy of 95.5%. Moreover, as a side effect of the triple compression also Q_2 is predicted with 76.6% accuracy.

The insights found for the triple compression experiments were confirmed on 4 times JPEG compressed images (Figure 2.14). Even in this scenario, if high QF are employed in the third compression (e.g., 90, 95, 100) Q_2 factors are actually predicted in a similar way of what was described before. Besides, if both QF_3 and QF_2 are high, Q_1 elements could be estimated, confirming how the method in each case obtains information about previous compressions.



(a)

(b)

(c)

Figure 2.14: Accuracy of the proposed method on JPEG 4-compressed images employing all the combinations $QF_1, QF_2, QF_3 \in \{60, 65, 70, 75, 80, 85, 90, 95, 100\}$ and $QF_4 = 90$ considering QF_3 as ground truth (a). Further analysis have been conducted with $QF_3 \in \{90, 95, 100\}$ (low accuracy regions): (b) and (c) show the results employing QF_2 and QF_1 as ground truth respectively.

The proposed method estimates the strongest previous compression which is basically the behavior of most First Quantization Estimation (FQE) methods. For this reason, a comparison was made with [63] on triple compressed images considering Q_{m-1} as correct estimation. Figure 2.15 reports the accuracy in the $QF_3 = 90$ scenario showing how our method (left graph) maintains good result even in triple compression while [63] has a significant performance drop compared to double compression.

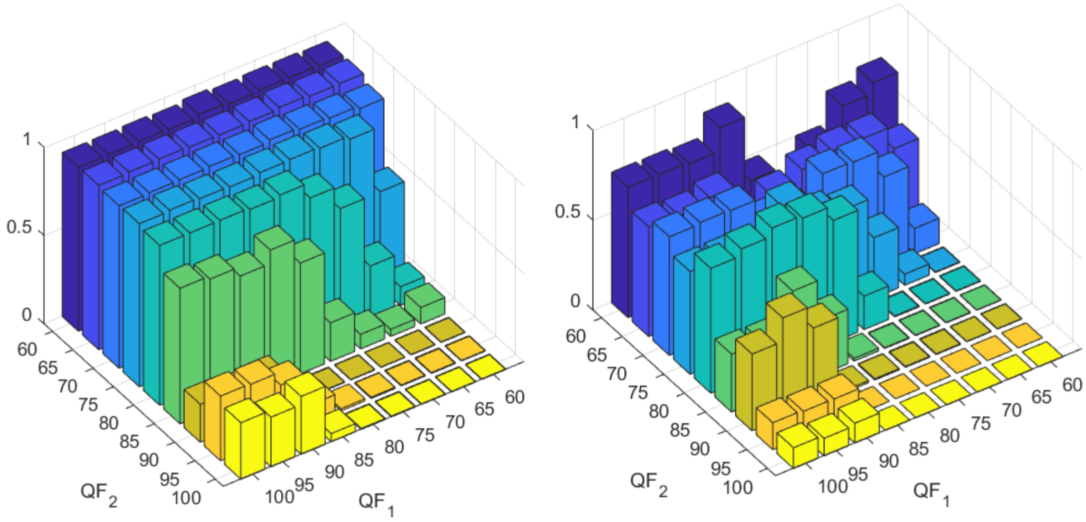


Figure 2.15: Accuracy of our method (left) and [63] (right) on JPEG triple compressed images employing all the combinations $QF_1, QF_2 \in \{60, 65, 70, 75, 80, 85, 90, 95, 100\}$ and $QF_3 = 90$ considering QF_2 as ground truth.

Recent works in literature demonstrate how different JPEG implementations could employ various Discrete Cosine Transform and mathematical operators to perform floating-point to integer conversion of DCT coefficients [76]. In order to further validate the proposed method, a cross JPEG implementation test was conducted considering two different libraries (Pillow and libjpeg-turbo) and 2 DCT configurations [8] to compress the input images and Pillow to simulate the double compression described in the pipeline. The test was performed using the same 8156 RAISE images cropped 64×64 and double compressed by means of the aforementioned JPEG implementations with $QF_1 = \{60, 65, 70, 75, 80, 85, 90, 95\}$ and $QF_2 = 90$. Results reported in Table 2.7 confirm the overall robustness of the proposed solution with respect to different JPEG implementations.

[8] https://github.com/libjpeg-turbo/libjpeg-turbo/

	60	65	70	75	80	85	90	95	**MEAN**
Pillow	0,76	0,77	0,78	0,82	0,82	0,75	0,23	0,38	**0,66**
libjpeg-turbo dct-int	0,76	0,76	0,77	0,80	0,80	0,74	0,23	0,38	**0,66**
libjpeg-turbo dct-float	0,75	0,75	0,76	0,80	0,80	0,72	0,23	0,39	**0,65**

Table 2.7: Accuracy obtained employing different JPEG implementations with $QF_2 = 90$. The columns (60,65,. . . ,95) represent the QF_1.

2.4 FQE by a smart data exploitation

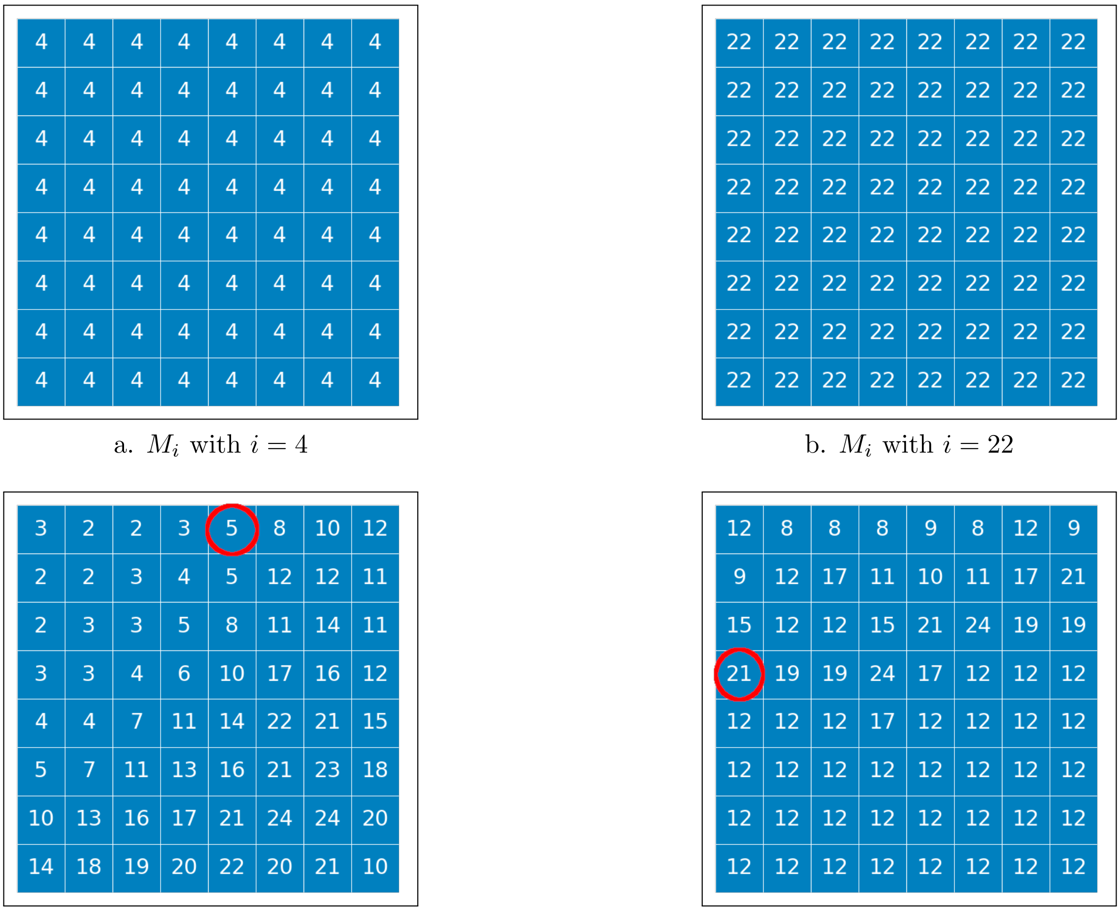
As discussed previously, most First Quantization Estimation methods in state of the art employ directly images as input. This choice, especially in Machine Learning methods, limits the usability of the built models to a specific patch size. Another limit of CNN-based works is the use of the second quantization matrix; in almost all cases the input for training phase has always data compressed with a standard second quantization matrix, producing models that satisfy only this condition.

To overcome those design limits, an hybrid solution based on both statistical analysis and machine learning was proposed (in the aligned scenario). The statistical side of the solution is represented by the pre-processing analysis of the image which exploit the statistical feature of the DCT distributions, while the machine learning is applied to predict the quantization factors. The use of DCT histograms as input (instead of the image), allows a patch size independence and an uncorrelated prediction from the input image quantization matrix. Moreover that choice permits the final user to make prediction also in presence of custom quantization matrices avoiding overfitting. In the following sections will be presented 2 different solutions based on the same statisical pre-processing, but with different machine learning methods in the prediction phase.

2.4.1 K-nearest neighbors (KNN) method

The technique involves the use of a proper dataset to compare against the query image for FQE. The first version and its extension have been published ([77, 78]).

The first part is then the generation of this reference dataset. Starting from the RAISE dataset [68] (8156 high-resolution uncompressed images) a 64×64 patch was extracted from the center of the original one. Considering every possible couple of constant matrix $\{M_i, M_j\}$ (Figures 2.16a, 2.16b) with $i, j \in \{1, 2, \ldots, q1_{max}\}$ as first and second quantization matrix,a double compression of the 64×64 image was



4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4

a. M_i with $i = 4$

22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22
22	22	22	22	22	22	22	22

b. M_i with $i = 22$

3	2	2	3	5	8	10	12
2	2	3	4	5	12	12	11
2	3	3	5	8	11	14	11
3	3	4	6	10	17	16	12
4	4	7	11	14	22	21	15
5	7	11	13	16	21	23	18
10	13	16	17	21	24	24	20
14	18	19	20	22	20	21	10

c. Standard matrix $QF = 90$

12	8	8	8	9	8	12	9
9	12	17	11	10	11	17	21
15	12	12	15	21	24	19	19
21	19	19	24	17	12	12	12
12	12	12	17	12	12	12	12
12	12	12	12	12	12	12	12
12	12	12	12	12	12	12	12
12	12	12	12	12	12	12	12

d. Photoshop matrix $QF = 5$

Figure 2.16: Example of the constant matrix M_i with $i = 4$ (a) and $i = 22$ (b), standard quantization matrix with $QF = 90$ (c) and custom quantization matrix extracted from Photoshop with quality 5; Photoshop defines its custom table with quality factor in range [1,12]. The red circle in (c) and (d) defines the maximum value of the matrix between the first 15 coefficients.

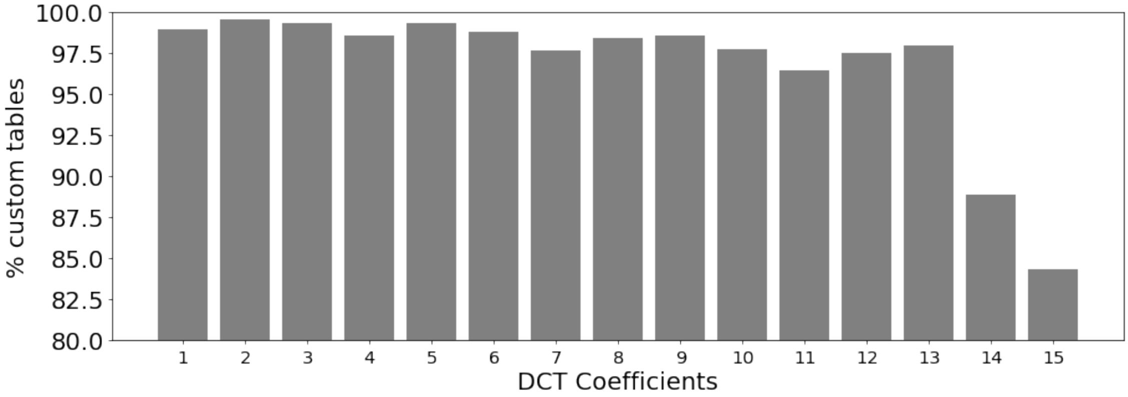


Figure 2.17: Percentages of custom matrices in Park's dataset [73] with quantization factors lower than 23 for the first 15 coefficients.

carried out. The choice of $q1_{max} = 22$ has been done taking into account realistic scenarios as detailed in the following. State-of-the-art often considered $QF \geq 55$ as the worst case to analyze for the task of FQE; considering that the best methods reach to estimate the first $k = 15$ DCT coefficients, the value of 15-th quantization factor of the matrix $QF = 55$ is 22. Moreover, considering the collection of matrices employed in real-case-scenario described in Park et al. [73] it means considering the 81% of them (873 on 1070) (Fig. 2.17). Accordingly, $8156 \times 22 \times 22 = 3.947.504$ double compressed images were generated, which represent the dataset that will be employed to do the comparison with the query image. The employing of constant matrices M_i as quantization matrices in double compression was done in order to safely break the correlation with real quantization matrices (e.g., standard ones). This way to compose the dataset comparison makes the approach generalizable w.r.t. any JPEG double compressed image (in the aligned scenario).

The comparison between the reference dataset and the query image (JPEG double compressed image) will be performed employing the DCT coefficient distributions h_i, with $i \in 1, 2, ..., q1_{max}$. Hence, the first $k = 15$ distributions h_i were calculated for each double compressed image, and then clustered in sub-datasets labelled by the couple $\{q1, q2\}$ representing first and second quantization factor employed. The usefulness of the Laplacian distribution (2.3) was described during the years ([79]); for this reason the distributions h_i of ACs coefficients were fitted by considering the following equation:

$$f(x) = \frac{1}{2\beta} \exp\left(\frac{-|x-\mu|}{\beta}\right) \tag{2.3}$$

where β and μ are obtained through MLE (Maximum Likelihood Estimation) closed form solution.

β values were employed for sub-datasets sorting of the AC distributions (h_i with $i > 1$), while the median value m to sort DC distributions h_1. To sum up, the reference dataset after sorting is composed as follows:

- DC_{dset}: DC distributions split for every possible couple $\{q1, q2\}$ sub-datasets (sorted by m);
- AC_{dset}: AC distributions split for every possible couple $\{q1, q2\}$ sub-datasets (sorted by β).

The organization of the data exploiting the model described in (2.3) permitted the reduction of the computational time of the proposed solution, in fact β and m will be employed to individuate the subdataset of the reference ones to select for the comparison with the query image. It is easy to note that the performance of the method depends on the size of the reference dataset: the bigger will be the amount of data, the better will be the estimation. The overall dataset sizes in our experiments have been selected accordingly with the results and the storage possibilities.

Given a double compressed image I'' as query, the estimation of the first k quantization factors $\{q1_1, q1_2, \ldots, q1_k\}$ of Q_1 could be summarized as in Algorithm 2 and Figure 2.18.

The following steps describe the estimation of every $q1_i$ with $i \in \{1, 2, \ldots, k\}$:

1. extract h_i from I'' (employing LibJpeg C library [9]).

[9] https://github.com/LuaDist/libjpeg

Algorithm 2 FQE method through KNN

```
**Input:** double compressed image $I''$
**Output:** $\{q1_1, q1_2, \ldots, q1_k\}$
*Initialization* : $k$, $q1_{max}$
**for** $i = 1$ to $k$ **do**
    $h_i$ : (empirical) distribution of $i$-th DCT coefficient
    **if** $(i = 1)$ **then**
        $D : DC_{dset}$
        $m$ : median value of $h_i$
    **else**
        $D : AC_{dset}$
        $\beta$ : $\beta$ fitted on Laplacian $h_i$
    **end if**
    $q2_i$ : quantization factor of $Q_2$ for $i$-th DCT
    **for** $j = 1$ to $q1_{max}$ **do**
        $D_{j,q2_i}$ : sub-dataset $(q1, q2)$ with $q1 = j$, $q2 = q2_i$
        $D_{j,q2_i}(m, \beta)$ : sub-range with most similar $m, \beta$
        $d_{i,j}$ : lower $\chi^2$ distance between $h_i$ and $D_{j,q2_i}$
    **end for**
    $q1_i : \arg\min_{\{d_{i,j}\}}$, $j \in \{1, 2, \ldots, q1_{max}\}$
**end for**
regularize($\{q1_1, q1_2, \ldots, q1_k\}$)
**return** $\{q1_1, q1_2, \ldots, q1_k\}$
```

2. fit h_i to the Laplacian distribution in order to extract β (if $i > 1$) or compute the median value m (if $i = 1$).

3. use β (or m) to seek the most similar item in the reference dataset and then to individuate the range of candidates inside it.

4. find the most similar distribution between the candidates.

Note that step 1 permits avoiding further errors (truncation and rounding) and that the usage of β (if $i > 1$) and m (if $i = 1$) makes the time to individuate the most similar distribution between the candidates constant.

The knowledge of Q_2 and $q2_i$ allows to select only some sub-datasets. For each sub-datset $\{q_j, q2_i\}$ with $j \in \{1, 2, \ldots, q1_{max}\}$ a range of elements $D_{j,q2_i}(m, \beta)$ is

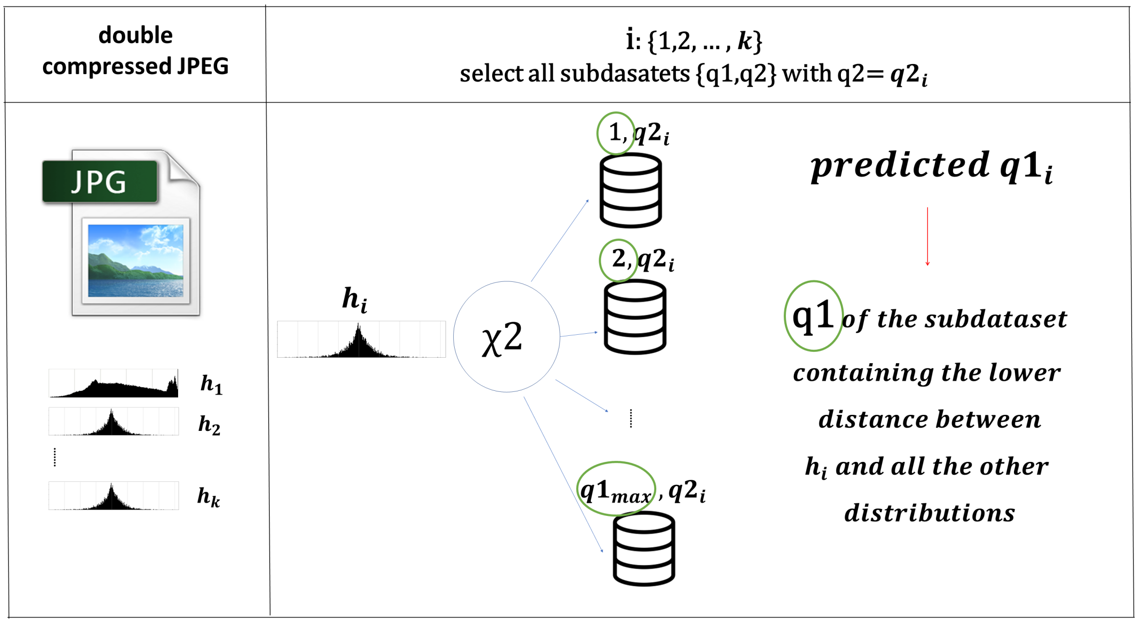

double compressed JPEG
i: {1,2, ... , k}
select all subdasatets {q1,q2} with q2= q2_i
JPG
h_1
h_2
h_k
h_i
χ2
1, q2_i
2, q2_i
q1_max, q2_i
predicted q1_i
q1 of the subdataset containing the lower distance between h_i and all the other distributions

Figure 2.18: A schematic representation of the proposed FQE method. The pipeline starts with a JPEG double compressed image. Each DCT histogram h_i is then compared with proper subdatasets computing χ^2 distance. Most similar histogram for each element is chosen for the FQE task.

selected through the similarity of β for AC_{dset} and m for DC_{dset}. The selected elements were compared with h_i using χ^2 distance:

$$\chi^2(x, y) = \sum_{i=1}^{m} (x_i - y_i)^2/(x_i + y_i) \tag{2.4}$$

where x and y are the distributions to compare. For each sub-dataset $D_{j,q2_i}$ the lowest distance $d_{i,j}$ is selected, obtaining $q1_{max}$ distances. The minimum distance individuates a sub-dataset, related to a specific $q1$ that represents the first quantization factor to estimate for the current i.

The obtained distances $d_{i,j}$ does not obtain always a strong minimum. Analyzing those specific cases, it is possible to note that it happens when the histogram h_i does not contains enough information to discriminate among the $q1_i$ values to be estimated. To overcome such limit an exploitation of the data related to neighboring DCT coefficients has been considered. In particular, as reported often in the state-of-the-art, the hypothesis that consecutive quantization factors in zig-zag order have similar values has been exploited. Specifically, with $k = 15$ and taking into account

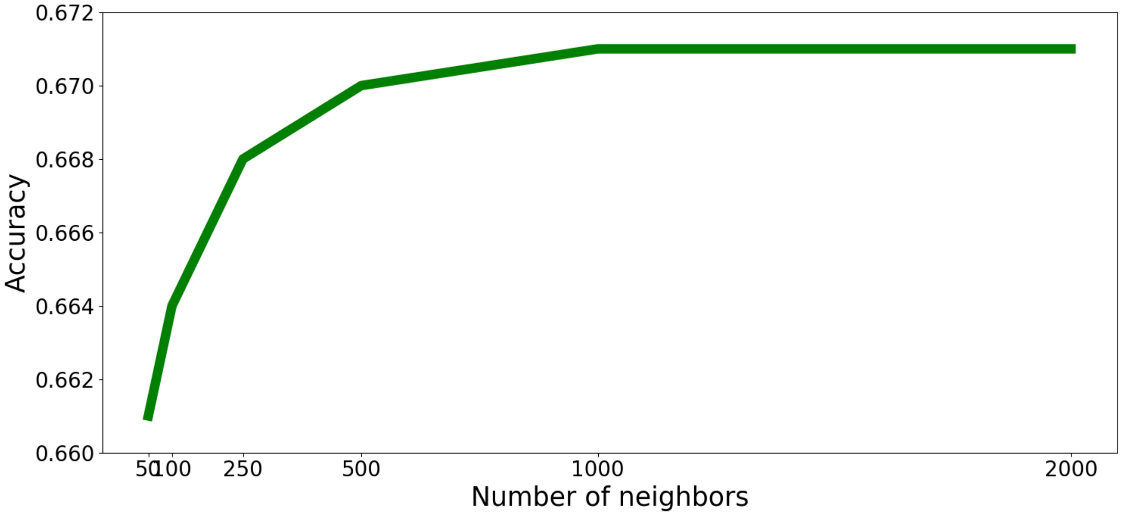


Figure 2.19: Accuracy of the proposed approach w.r.t. the number of neighbors considered for comparison.

three consecutive $q1$, 13 triplets ($q1_{i-1}$, $q1_i$, $q1_{i+1}$, $i = 2, \ldots, 14$) can be found.

A single triplet ($q1_{i-1}$, $q1_i$, $q1_{i+1}$) can be then estimated through a cost function in which all possible $q1$ combinations (i.e., $22 \times 22 \times 22$ with $q1_{max} = 22$) are considered. The cost function C is designed as the weighted average of a data term (C_{data}) and a regularization term (C_{reg}):

$$C = wC_{data} + (1 - w)C_{reg} \tag{2.5}$$

where $w \in [0, 1]$, C_{data} is the normalized sum of the three distances $d_{i,j}$, and C_{reg} is the regularization term introduced to reduce the difference among neighboring $q1$ values. Additional details related to the design of C_{reg} will be presented during method explanation.

Parameter setting

In order to find the right parameters for the method described in Section 2.4.1 a validation dataset D_V was employed, composed of 8156 64×64, cropped at random positions from the RAISE [68] dataset.

As previously described, given the second quantization factor $q2$ (from the meta-data) the reference sub-datasets selected are $q1_{max}$, in which a sort by m and β values is carried out to reduce the computational complexity. The sorting allows to fix an

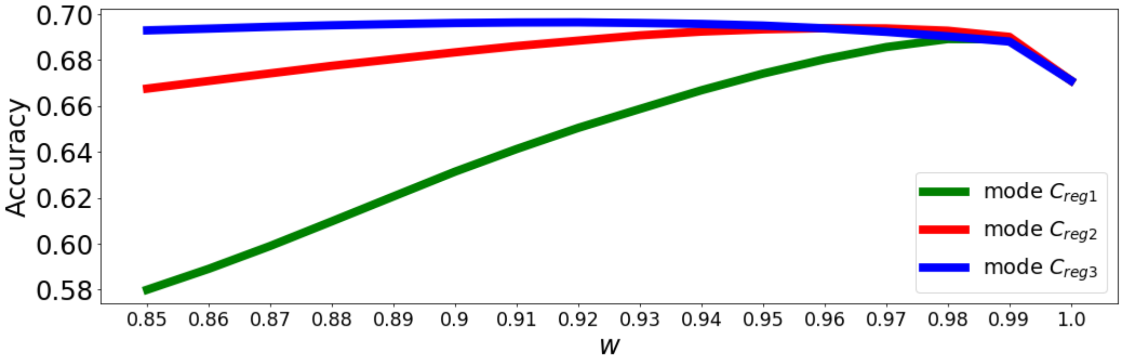


Figure 2.20: Comparison between the modes of the accuracies obtained for different values of the regularization parameter w employing the three equations (2.6), (2.7), (2.8) respectively.

exact number of distributions h_i to be taken into account making the computational time constant and not dependent on the dataset size. The right number of comparisons was chosen after tested sub-range of 50, 100, 250, 500, 1000, 2000 elements. The results of Figure 2.19 show as the value of 1000 is a good trade-off between the overall computational cost and the accuracy w.r.t. the full search solution.

Sometimes, the described histograms h_i do not contain enough information, then a regularization function was employed in order to cope with this problem. As mentioned in Section 2.4.1, the proposed approach estimates triplets of nearest $q1_i$ and performs a regularization based on 2.10. In order to choose the right C_{reg} three different functions have been analyzed:

$$C_{reg1} = \frac{|c_i - c_{i-1}| + |c_i - c_{i+1}|}{2} \tag{2.6}$$

$$C_{reg2} = \frac{|c_i - c_{i-1}| + |c_i - c_{i+1}|}{2\sqrt{c_i}} \tag{2.7}$$

$$C_{reg3} = \frac{|c_i - c_{i-1}| + |c_i - c_{i+1}|}{2c_i} \tag{2.8}$$

where c_{i-1}, c_i, c_{i+1} are consecutive quantization factors candidates in zig-zag order. Every $q1_i$ is estimated multiple times: as example $q1_3$ is estimated in three different triplets: $(q1_1, q1_2, q1_3)$, $(q1_2, q1_3, q1_4)$ and $(q1_3, q1_4, q1_5)$. As reported in Fig. 2.20 the modes of multiple estimations related to eq.2.6,2.7,2.8 were studied:

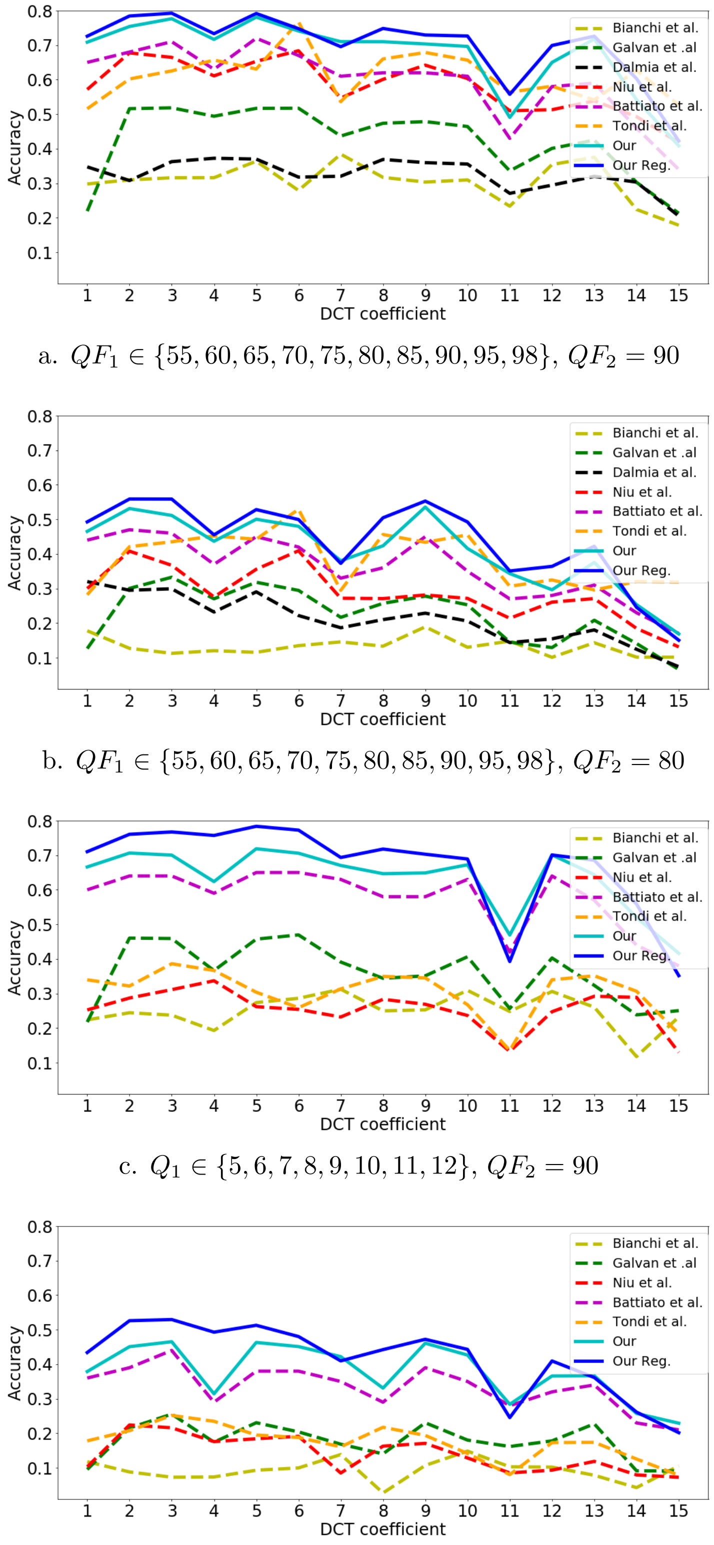
the last (2.8) was chosen with $w = 0.92$.

Experimental Results

QF_1	$QF_2 = 90$							
	Our	*Our Reg.*	*[33]*	*[53]*	*[54]*	*[63]*	*[64]*	*[65]*
55	0.76	**0.77**	0.53	0.52	0.45	0.00	0.00	0.62
60	**0.82**	**0.82**	0.53	0.56	0.47	0.64	0.53	0.66
65	0.79	**0.81**	0.54	0.57	0.49	0.54	**0.81**	0.68
70	**0.85**	**0.85**	0.43	0.57	0.51	0.66	0.66	0.70
75	0.83	0.85	0.41	0.63	0.53	0.77	**0.93**	0.75
80	0.81	**0.83**	0.29	0.61	0.45	0.81	0.67	0.75
85	0.78	0.85	0.14	0.74	0.36	0.81	**0.88**	0.72
90	**0.30**	0.24	0.00	0.00	0.00	0.02	0.02	0.23
95	0.44	0.52	0.11	0.00	0.00	**0.78**	0.68	0.40
98	0.49	0.57	0.00	0.00	0.00	0.76	**0.91**	0.41
MEAN	0.69	**0.71**	0.30	0.42	0.33	0.58	0.61	0.59
QF_1	$QF_2 = 80$							
	Our	*Our Reg.*	*[33]*	*[53]*	*[54]*	*[63]*	*[64]*	*[65]*
55	0.55	**0.58**	0.36	0.37	0.37	0.24	0.52	0.49
60	0.55	**0.60**	0.27	0.37	0.38	0.50	0.31	0.47
65	**0.68**	0.65	0.19	0.41	0.43	0.31	0.38	0.52
70	0.67	**0.75**	0.19	0.50	0.49	0.50	0.70	0.66
75	0.48	0.56	0.07	0.56	0.45	0.15	**0.58**	0.47
80	**0.12**	0.11	0.00	0.00	0.00	0.00	0.04	0.04
85	0.28	**0.34**	0.19	0.00	0.00	0.04	0.14	0.31
90	0.16	0.19	0.06	0.00	0.00	**0.48**	0.37	0.21
95	0.27	0.30	0.00	0.00	0.00	**0.95**	0.37	0.19
98	0.42	0.42	0.01	0.00	0.00	0.21	**0.48**	0.19
MEAN	0.42	**0.45**	0.13	0.22	0.21	0.28	0.39	0.36

Table 2.8: Accuracies obtained by the proposed approach compared to Bianchi et al. [33], Galvan et al. [53], Dalmia et al. [54], Niu et al. [63], Tondi et al. [64] and Battiato et al. [65] with different combinations of QF_1/QF_2, considering standard quantization tables. To note that Battiato et al. [65] is the method described in Section 2.3.

State-of-the-art showed abundantly as the difficulty to estimate a specific quantization factor is related to its position: High frequencies are used to be quantizated heavily and after certain values the recovered value after dequantization is always zero (e.g., 'dead zone' quantization). Although some state-of-the-art papers analyzed a lower value, the setting of $k = 15$ as the number of quantization factor to estimate has became almost a standard for this task, and it was adopted also in this approach.



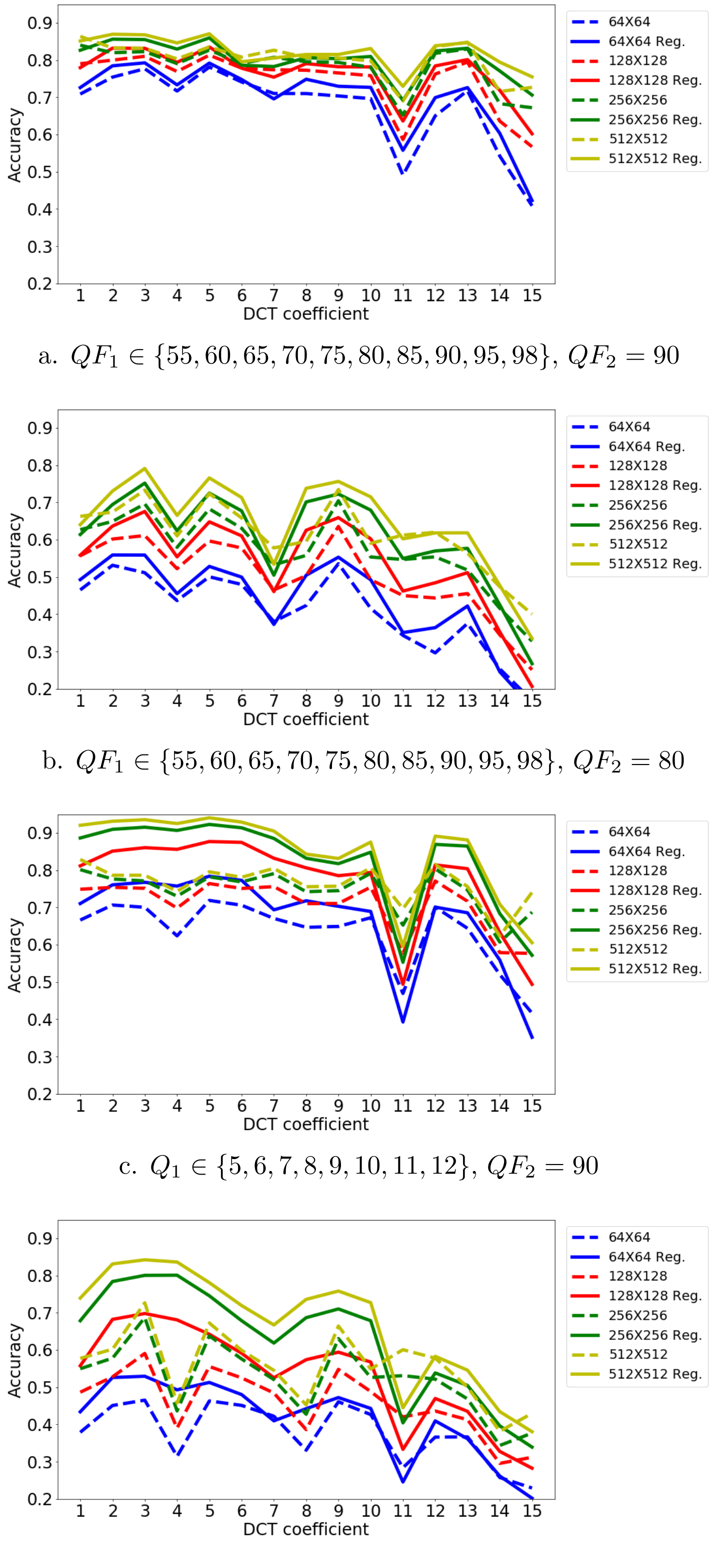
a. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$

b. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$

c. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$

d. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

Figure 2.21: Accuracies of the same methods described in Table 2.8 and 2.9 at varying the quantization factors $q1_i$ to be predicted. The values are averaged over all the QF_1/Q_1. Comparisons have been done with Bianchi et al. [33], Galvan et al. [53], Niu et al. [63],Tondi et al. [64] and Battiato et al. [65]. To note that Battiato et al. [65] is the method described in Section 2.3.



a. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$

b. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$

c. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$

d. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

Figure 2.22: Accuracies of the proposed method at varying training/test patch size. The values are averaged over all QF_1/Q_1.

PS	$QF_2 = 90$						
	Our	*Our Reg.*	*[33]*	*[53]*	*[63]*	*[64]*	*[65]*
5	**0.80**	0.78	0.56	0.58	0.05	0.10	0.69
6	**0.82**	**0.82**	0.46	0.60	0.07	0.13	0.75
7	**0.83**	**0.83**	0.41	0.58	0.07	0.09	0.70
8	**0.81**	**0.81**	0.25	0.65	0.10	0.13	0.74
9	0.55	**0.61**	0.02	0.47	0.02	0.05	0.47
10	0.42	**0.50**	0.19	0.00	0.25	0.33	0.43
11	0.45	0.52	0.04	0.00	0.69	**0.75**	0.40
12	0.49	0.57	0.04	0.00	0.75	**0.85**	0.41
MEAN	0.64	**0.68**	0.25	0.36	0.25	0.30	0.58
PS	$QF_2 = 80$						
	Our	*Our Reg.*	*[33]*	*[53]*	*[63]*	*[64]*	*[65]*
5	0.65	**0.68**	0.26	0.46	0.07	0.08	0.57
6	0.42	**0.54**	0.05	0.41	0.02	0.06	0.42
7	0.62	**0.68**	0.15	0.48	0.08	0.10	0.56
8	0.19	**0.22**	0.03	0.03	0.01	0.04	0.18
9	0.26	**0.28**	0.19	0.00	0.07	0.13	**0.28**
10	0.15	0.20	0.00	0.00	**0.40**	0.16	0.17
11	0.37	**0.38**	0.01	0.00	0.24	**0.38**	0.23
12	0.42	0.42	0.01	0.00	0.21	**0.44**	0.26
MEAN	0.39	**0.42**	0.09	0.18	0.17	0.14	0.33

Table 2.9: Accuracies obtained by the proposed approach compared to Bianchi et al. [33], Galvan et al. [53], Niu et al. [63], Tondi et al. [64] and Battiato et al. [65] employing custom tables for first compression. The column PS refers to custom tables used by Photoshop.

The goodness of the method was demonstrated through the comparison with different styles approaches: statistical analysis (mathods Bianchi et al. [33], Galvan et al. [53], Dalmia et al. [54], Battiato et al. [65]) and Machine Learning based methods (Niu et al. [63] and Tondi et al. [64]) were considered. For each one of the methods the tests were done with the same datasets and employing the public implementations provided by the authors.

4 dataset were generated to conduct the tests: a random (not centered) 64 × 64 patch was cropped from every RAISE [68] and then compressed two times with the following rules:

1. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$
2. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$
3. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$
4. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

where $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$ of 3) and 4) are referred to Photoshop's quantization matrices (CC version 20.0.4). Dalmia et al. [54] make assumptions about standard tables employed in the first compression in its implementations and so was excluded in the tests with Photoshop's custom tables (datasets 3 and 4). All the other methods were tested with the 4 datasets described above as input.

Tables 2.8, 2.9 and Fig. 2.31 show the results: our approach outperforms state-of-the-art methods in almost all scenarios (values close to 0 are due to the assumptions of some methods, e.g., $QF_1 < QF_2$). It is important to note that Table 2.9, where the custom quantization matrices were employed, demonstrates the absence of correlation with a particular quantization table and that the regularization step improves the results in almost all cases.

To further confirm the effectiveness of the proposed solution, additional experiments at varying patch size have been performed. The same conditions of tests already seen before, but with more informative patches have been considered. Specifically, other three double compressed training (or reference) datasets from RAISE with $128 \times 128, 256 \times 256, 512 \times 512$ patches have been generated. In the same way test datasets with each patch sizes (128×128, 256×256 and 512×512) with standard and custom tables (Photoshop) for Q_1 and $QF_2 \in \{80, 90\}$ have also been created.

In Fig. 2.22 it is easy to understand the trend of the results: due to the higher amount of information, bigger patches achieve higher accuracy in all the considered scenarios. The correlation between informative content of an image and estimation accuracy confirms the studies conducted in Section 2.3.

Method	**Test Dataset**	**Low/Low**	**Low/Mid**	**Low/High**	**Mid/Low**	**Mid/Mid**	**Mid/High**	**High/Low**	**High/Mid**	**High/High**	**Mean**
Our	RAISE [68] 64×64	0.25	0.47	0.79	0.17	0.32	0.82	**0.27**	0.31	0.70	0.46
Our Reg.		**0.30**	**0.53**	**0.81**	**0.22**	**0.37**	**0.84**	0.25	**0.33**	**0.75**	**0.49**
Our	UCID [72] 64×64	0.33	0.63	0.93	0.20	0.39	0.90	**0.15**	0.21	0.66	0.49
Our Reg.		**0.36**	**0.65**	**0.96**	**0.23**	**0.42**	**0.91**	0.13	**0.23**	**0.73**	**0.51**
Our	RAISE [68] 128×128	0.36	**0.60**	0.85	0.29	0.44	0.87	**0.25**	0.32	0.74	0.52
Our Reg.		**0.41**	0.55	**0.88**	**0.34**	**0.48**	**0.89**	**0.25**	**0.38**	**0.79**	**0.55**
Our	UCID [72] 128×128	0.47	0.76	0.96	0.31	0.49	**0.94**	0.18	0.29	0.74	0.57
Our Reg		**0.50**	**0.79**	**0.96**	**0.35**	**0.51**	0.93	**0.20**	**0.34**	**0.79**	**0.60**
Our	RAISE [68] 256×256	0.45	0.69	0.88	0.38	0.52	0.89	0.25	0.36	0.77	0.58
Our Reg.		**0.49**	**0.73**	**0.90**	**0.40**	**0.55**	**0.90**	**0.30**	**0.45**	**0.82**	**0.62**
Our	UCID [72] 256×256	0.56	0.83	**0.98**	0.44	0.57	**0.96**	0.23	0.34	0.77	0.63
Our Reg.		**0.60**	**0.85**	0.97	**0.48**	**0.60**	**0.96**	**0.28**	**0.43**	**0.82**	**0.67**
Our	RAISE [68] 512×512	**0.50**	0.74	0.91	0.44	0.57	0.91	0.26	0.38	0.77	0.61
Our Reg.		**0.50**	**0.78**	**0.92**	**0.48**	**0.59**	**0.92**	**0.32**	**0.48**	**0.83**	**0.65**
Our	UCID [72] Full size	0.63	0.86	**0.98**	0.51	**0.62**	**0.96**	0.27	0.39	0.80	0.66
Our Reg.		**0.67**	**0.87**	0.97	**0.56**	**0.62**	**0.96**	**0.37**	**0.49**	**0.85**	**0.71**

Table 2.10: Accuracies of the proposed approach for generalizing property demonstration (Our Reg. denotes the regularized version). It is worth noting that, given a patch size $N \times N$, the same dataset was employed as reference (i.e., RAISE $N \times N$) for both input (i.e., RAISE $N \times N$, UCID $N \times N$).

Recently, Park et al. in [73] collected a dataset of JPEG quantization matrices employed in real scenarios. The collection is made up of 1170 different quantization matrices, including the 100 standard ones. In order to demonstrate a further generalization capability the collection was exploited: as said before the reference datasets created concern quantization matrix with $q1_{max} = 22$. For this reason, after the download of the collection, only the collection's matrices with $q1_i \leq q1_{max} = 22$ and $i \in \{1, 2, \dots, 15\}$ (see Section 2.4.1) have been selected; moreover we sorted them by the average of the first 15 quantization factors and then split into three sets of 291 elements (Low, Mid, High). These sets of tables are then employed to create 9 combinations of double compressions.

We carried out the tests employing four different patch sizes (64×64, 128×128, 256×256, 512×512) and selecting randomly the quantization matrices Q_1 and Q_2

	60	65	70	75	80	85	90	95	**MEAN**
Pillow	0,86	0,83	0,88	0,87	0,86	0,79	0,31	0,42	**0,73**
libjpeg-turbo	0,85	0,82	0,87	0,85	0,83	0,77	0,32	0,42	**0,72**

Table 2.11: Accuracy obtained employing different DCT implementations for JPEG compression. It is worth noting that the different implementation are referred to the test set; the reference dataset is the same employed in the previous tests (i.e., RAISE double compressed images with DCT Pillow implementation).

to perform the double compression from the 291 available in the corresponding set (Low, Mid, High). The results reported in Table 2.10 clearly show that the good accuracies reached before are maintained also in this challenging scenario demonstrating to achieve same results even when different datasets are employed (e.g., RAISE as reference and UCID as test dataset).

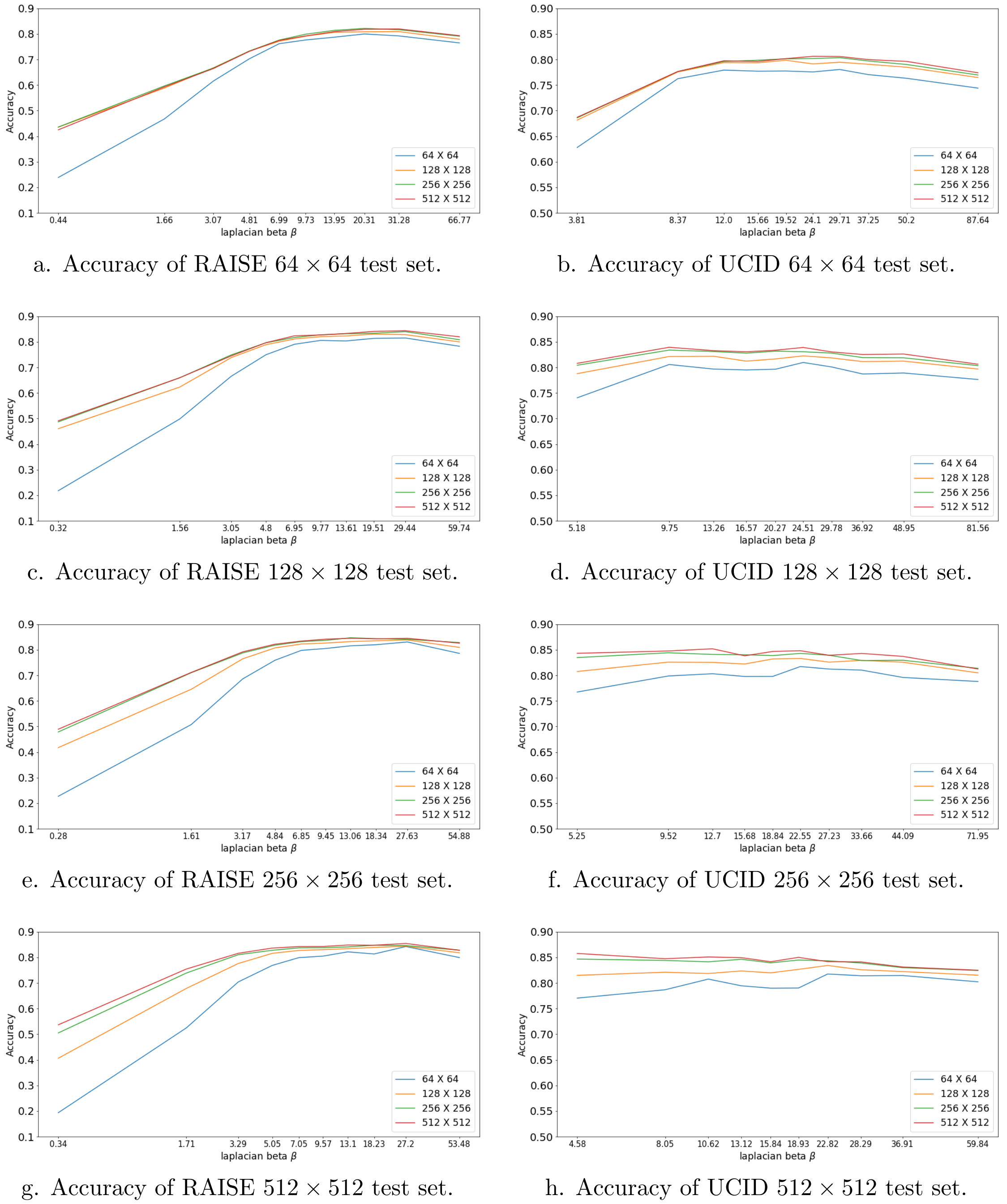
Once more, the results underline the importance of the content: the better results with UCID [72] dataset are explainable with the greater quantity of information (the DCT overall energy) inside it, in fact, a 256 × 256 patch extracted from a UCID [72] image (512 × 384) has a quantity of information greater than a 256 × 256 patch extracted from a RAISE [68] image (≈ 5000 × 3000).

A final test was done for the proposed approach. Recently,the research community is studying the effect of different Discrete Cosine Transform implementations in the JPEG image generation [76]. In order to validate the proposed method, also in presence of different JPEG implementations, was conducted a test employing Pillow and libjpeg-turbo[10] to compress two times the images. The test was done testing a dataset composed by 2000 RAISE images cropped 64 × 64 and compressed two times with the aforementioned JPEG implementations. Also in this test,the results in Table 2.11 confirmed the overall robustness of the method also w.r.t different DCT implementations.

Deepening

All the experiments performed previously were done considering a specific patch size $N \times N$ for each scenario, where both reference and test sets have been generated starting from patches of the same size. As demonstrated, bigger patch sizes, due to the higher information content, provide better results (see Fig. 2.22 and Table 2.10).

[10] https://github.com/libjpeg-turbo/libjpeg-turbo/



a. Accuracy of RAISE 64 × 64 test set.

b. Accuracy of UCID 64 × 64 test set.

c. Accuracy of RAISE 128 × 128 test set.

d. Accuracy of UCID 128 × 128 test set.

e. Accuracy of RAISE 256 × 256 test set.

f. Accuracy of UCID 256 × 256 test set.

g. Accuracy of RAISE 512 × 512 test set.

h. Accuracy of UCID 512 × 512 test set.

Figure 2.23: Accuracy of tests conducted on different test sets grouped and sorted by Laplacian β, shown on a logarithmic scale. It is worth noting that, given a patch size $N \times N$, the same dataset was employed as reference (i.e., RAISE $N \times N$) for both input (i.e., RAISE $N \times N$, UCID $N \times N$).

The next step was to understand if it is possible to exploit information contained in patches of different size with respect to the analyzed ones so that, a real application could refer to a unique reference dataset. To do that cross patch size tests have been conducted considering 8 datasets built from RAISE [68] and UCID [72]. In particular for every couple of patch size $64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512$ a test was conducted, employing for each one 1000 random images as query images, compressed two times with $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$. The generated images have been the compared with the 4 reference datasets employed in previous sections. Tables 2.12 and 2.13 demonstrate how training datasets built from higher patch size work better almost always, due to the higher information content. To understand how these improvements are related to the information content, the accuracies of AC coefficient estimation in all the considered scenarios at varying β (eq. (2.3)) was studied.

Training Test	64	128	256	512
64	0.677	0.699	**0.703**	0.699
128	0.698	0.744	0.758	**0.760**
256	0.705	0.758	0.780	**0.783**
512	0.710	0.768	0.792	**0.801**

Table 2.12: Accuracies obtained for every couple test/training with RAISE as test dataset.

Although exploiting training datasets with bigger patch sizes always provides better results (see Fig. 2.23), the gain in terms of accuracy is higher when low β values are considered (i.e., lower informative patches). This behavior is more evident in the experiments conducted on RAISE dataset (w.r.t. UCID) due to the different resolution of the original images used to extract patches. AC distributions with low β values have almost all bins close to zero and it is rather difficult to discriminate among ones obtained with different first quantization factors. This is even worse when a limited number of elements are used to build these histograms: histograms generated from 64×64 and 512×512 patches have been obtained considering 64

Test \ Training	64	128	256	512
64	0.754	0.770	0.775	**0.777**
128	0.789	0.812	0.824	**0.829**
256	0.799	0.824	0.837	**0.843**
512	0.799	0.824	0.842	**0.846**

Table 2.13: Accuracies obtained for every couple test/training with UCID as test dataset.

and 4096 elements respectively. Higher patch sizes allow then, especially when the information content is limited, to better discriminate similar distributions.

2.4.2 Convolution Neural Network (CNN) Method

The excellent results obtained in Section 2.4.1 (Battiato et al.[78]) exploiting a simple k-nn ($k = 1$), suggested a further analysis of the method.

For each $q2 \in \{1, 2, \ldots, q1_{max}\}$, 2 reference datasets of empirical distributions were built, the first related to AC and the second to DC, due to their composition difference. $q1_{max}$ is the maximum value between the first k factors in zig-zag order in the quantization matrices: considering as matrices the figures 2.16c and 2.16d, and fixing $k = 15$, $q1_{max}$ values are 5 and 21 respectively (highlighted in the figures). To sum up, $2 \cdot q1_{max}$ models allowed us to deal with double compressed images with custom first and second compression matrices Q_1 and Q_2 whose quantization factors are lower than or equal to $q1_{max}$ in the first k positions (zig-zag order). This data organization allows us to switch from ${q1_{max}}^k$ possible combinations to $2 \cdot q1_{max}$; considering the parameters setting chosen in 2.4.1 ($k = 15$ and $q1_{max} = 22$) means that only $22 \cdot 2 = 44$ models have to be trained employing our strategy instead of 22^{15}. The described data organization strategy allows to deal with a large number of double quantization parameters maintaining a feasible workload, in terms of computational efforts.

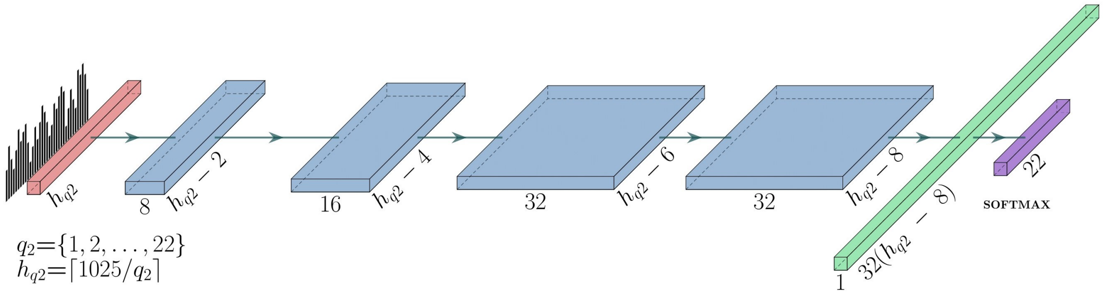


Figure 2.24: Parametric architecture representing the trained CNNs. The first layer represents the distribution of i-th DCT coefficient, compressed the second time with $q2$. The input distribution is then reduced in h_{q2} bins. The following four layers are 2D convolutions with a filter 1×3, batch normalization and ReLu activation function. The last two layers consist of a fully connected and a softmax layer with 22 elements.

Many state-of-the-art approaches based on machine learning train their models considering datasets related to a specific second quantization matrix Q_2, but in real applications, it is very likely to find double compressed images with Q_2 different than the one used in those studies. This would force the investigator to build datasets and to perform a training phase for each desired Q_2. In this Section the data organization described before was exploited to cope this problem, training CNNs that employ DCT distributions h_i obtained in the same way as previous section. Also in the method that will be described, RAISE [68] was considered in training phase. The choice derives from the desire to face a challenging condition: RAISE is composed of 8156 high-resolution uncompressed images that were captured in different scenes employing different cameras and the extraction of a 64×64 pixel images from high resolution images permit us to obtain distributions with a low level of information and then to train the network also in presence of critical cases.

The double compression phase was carried out extracting different $d \times d$ central patches from raw images and compressing the images employing proper combinations of constant matrices M_i (Fig. 2.16a, 2.16b) with $i \in \{1, 2, \ldots, q1_{max}\}$. In our tests, the value of $q1_{max}$ was set to 22 whereas 4 different patch sizes with $d \in \{64, 128, 256, 512\}$ were considered. Hence, the final number of JPEG double compressed images exploited to train the proposed solution was $8156 \times 22 \times 22 \times 4 =$

15.790.016. Moreover, starting from a double compressed image, $k = 15$ DCT distributions (1 DC and 14 AC) were extracted. Each distribution is then characterized by 4 parameters: patch size d, first quantization factor $q1$, second quantization factor $q2$ and specific DCT position k. Finally, to better organize this huge amount of data (i.e., $8156 \times 22 \times 22 \times 4 \times 15$), all the distributions were clustered according to the parameters d, $q2$ and DCT coefficient type (DC, AC) generating $4 \times 22 \times 2 = 176$ different sets. In the method described in Section 2.4.1 ([78]) we already demonstrated that a simple learning approach ($k - nn$ with $k = 1$) could obtain good results also with the variability of custom quantization matrices. Now, with a deep learning technique, a considerable improvement of the overall accuracy is expected whilst maintaining robustness and generalization properties. The following described method id currently under review to be published.

Given a double JPEG compressed image I'' the main aim of the proposed solution is the estimation of the k first quantization factors employed in the first compression. As already pointed out in Section 2.4.1, for each $q2$, two different DCT coefficient types (DC or AC), have to be taken into account, due to their difference in terms of statistical distribution. A DC-CNN and an AC-CNN for each possible value of $q2 \in \{1, 2, \ldots, q1_{max}\}$ were trained.

The CNN architecture is described in Fig. 2.24. For each $q2$ the input is the normalized DCT histograms with $h_{q2} = \lceil 1025/q2 \rceil$ bins. The size of the following layers depends on h_{q2}. Fig. 2.24 summarizes the neural network architecture layers: they consist of 2D convolutions carried out with 1×3 filters, a batch normalization and a ReLu activation (blue layers), a fully connected layer (green) and the softmax function with $q1_{max} = 22$ values as output layer. In training phase a Stochastic Gradient Descent (SGD) was employed as optimizer with starting learning rate 10^{-3} and momentum 9^{-1} and the categorical cross entropy was the loss function employed during a 15-epochs training run with batches of 512 images. Moreover a decay step on learning rate value was carried out, with the drop value described in Eq. 2.9.

$$lr^e = lr^0(d_r^{\lfloor (1+e)/ed \rfloor}) \tag{2.9}$$

where e is the epoch, lr^e is the learning rate of epoch e, lr^0 is the starting learning rate, $d_r = 0.2$ is the drop value and $ed = 3$ is the number of epochs for every drop.

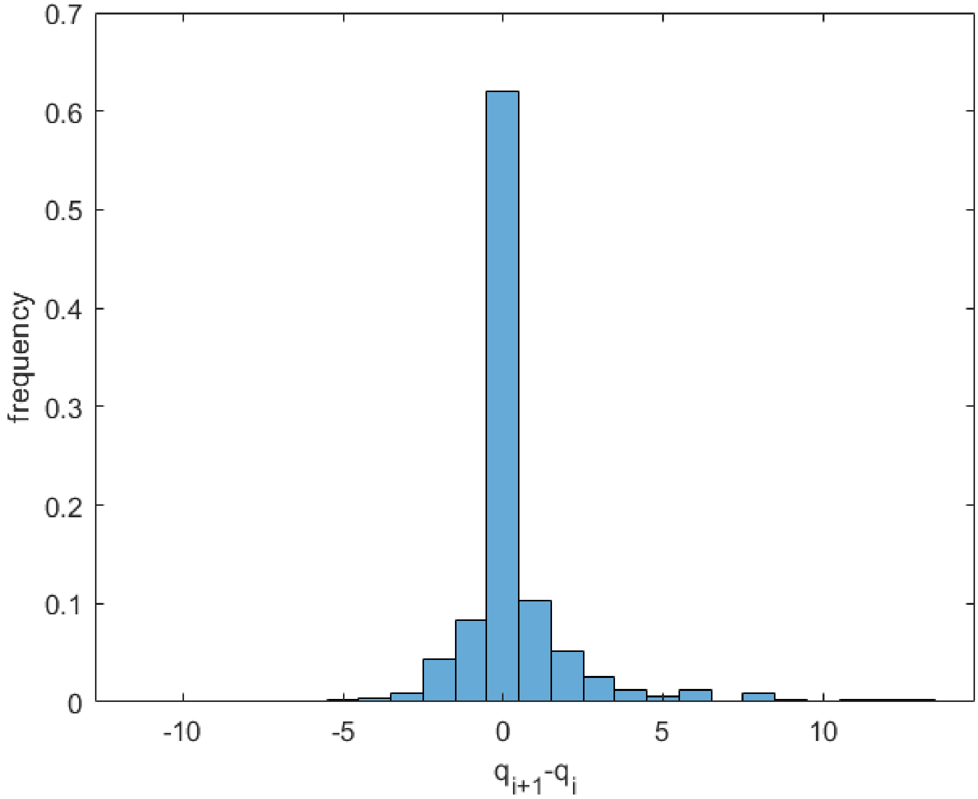


Figure 2.25: Empirical distribution of differences of consecutive quantization factors (zig-zag order) built from standard and custom matrices collected in [73] with $q_{1i} \leq q_{1max} = 22$ and $i \in 1, 2, \ldots, 15$.

As explained in previous sections, often the quantity of information contained in an input histogram could be not enough to estimate the related first quantization factor. This lack of information could be due to a specific $q1/q2$ (e.g., multiples) combination or an homogeneous region. To limit these issues, assumptions about neighboring element values in the quantization matrix can be exploited, as done in Section 2.4.1. In order to verify these assumptions empirically, an analysis on Park et al. [73] dataset was done. Considering only the matrices with $q_{1i} \leq q_{1max} = 22$ and $i \in 1, 2, \ldots, 15$, 919 tables (custom and standard) were selected and the empirical distribution of differences between consecutive quantization factors in zig-zag order were built. Fig. 2.25 shows how neighboring elements in the quantization matrices (zig-zag order) are usually associated to similar values (i.e., their difference is close to zero). Considering then a set of n consecutive first quantization factors to be

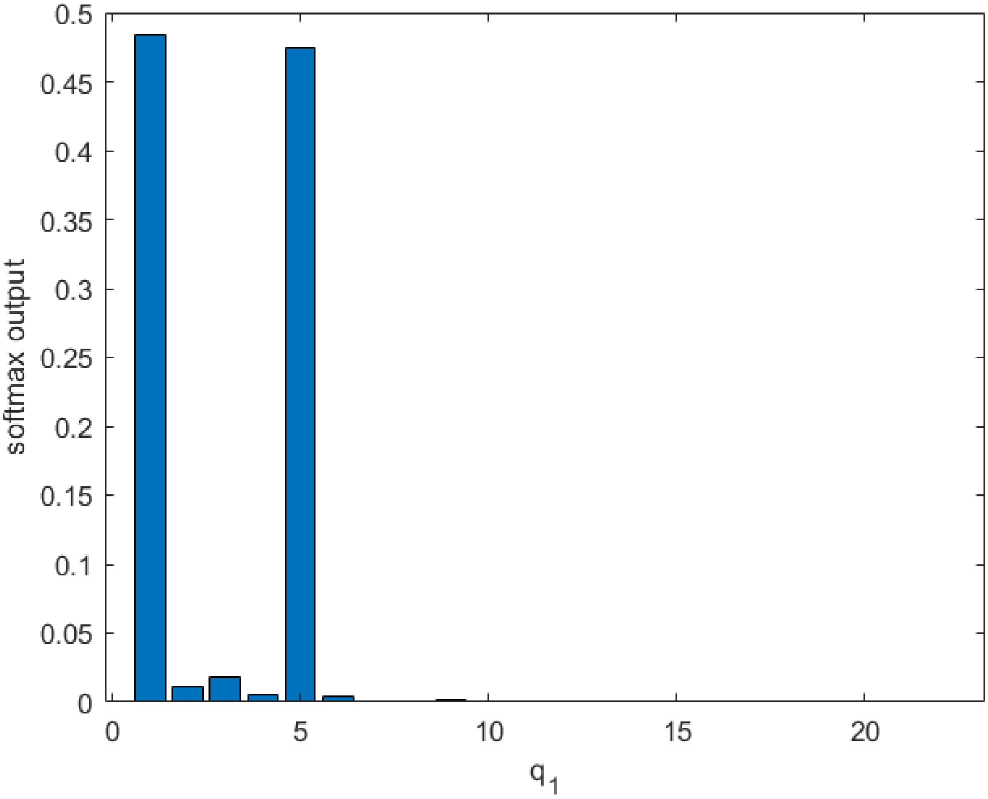


Figure 2.26: An example of softmax output provided by the proposed CNN considering an AC histogram obtained with $q1 = 5$ and $q2 = 5$.

estimated, a cost function C was designed as the weighted average of a data term (C_{data}) and a regularization term (C_{reg}):

$$C = w \cdot C_{data} + (1 - w) \cdot C_{reg} \tag{2.10}$$

where $w \in [0, 1]$, C_{data} is a cost term related to the goodness of the estimation of first quantization factors under analysis, and C_{reg} is a regularization term that tries to minimize differences among neighboring $q1$ values. In order to understand the right number of consecutive elements to be considered an analysis of the softmax outputs was performed. The information provided by the softmax output could be exploited to extract a limited set of candidates of $q1$: softmax is a $q1_{max}$ elements vector that describes the probabilities (they are all positives and sum to one) of every possible $q1$. Fig. 2.26 shows a softmax output computed by the proposed CNN with an histogram obtained from a double compressed 128×128 patch with $q1 = 5$ and

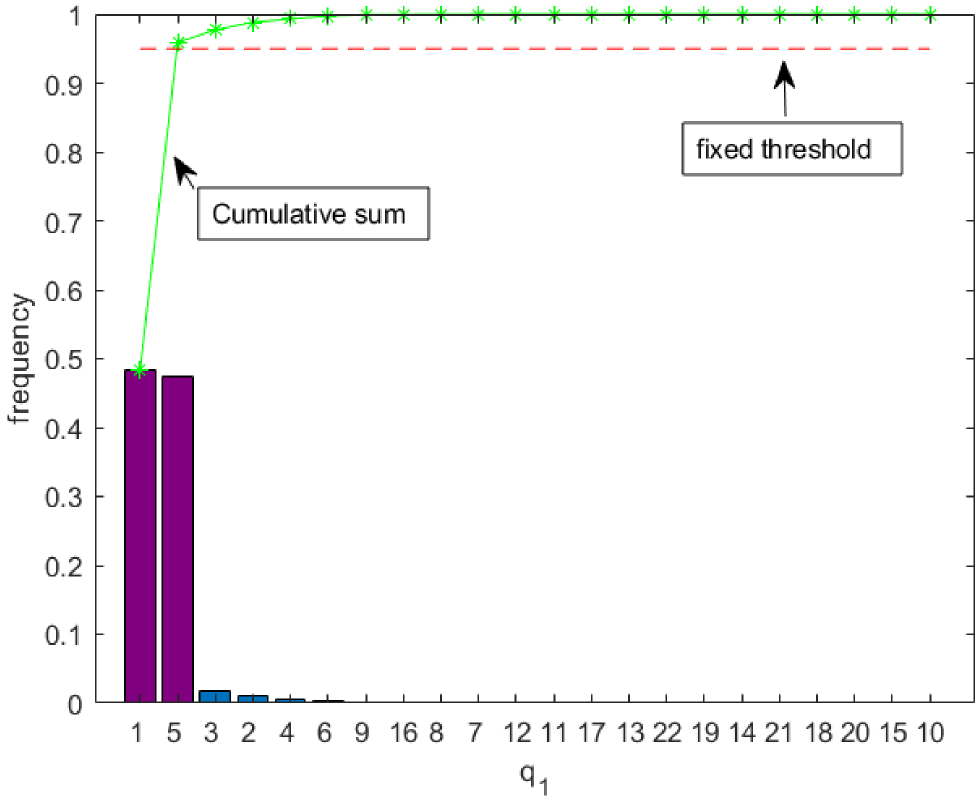


Figure 2.27: An example of $q1$ candidate selection from softmax output values provided by the proposed CNN considering the same input of Fig. 2.26. Softmax outputs p_i are sorted in descending order, the cumulative sum is computed (in green) and compared with a threshold th (in red). The first set of quantization factors whose cumulative sum is higher than th are considered as candidates (1 and 5).

$q2 = 5$ (challenging scenario). The network estimated a wrong value ($q1 = 1$) but the score associated to the right value $q1 = 5$ is similar with the estimated one: the probability associated to the event $q1 = 1$ is 0.484 whereas the joint probability related to $q1 = 1$ or $q1 = 5$ is 0.959. A set of first quantization factors can be then selected to achieve a satisfactory probability.

This behavior can be then exploited: the softmax output could be used as a reliability index of the estimation: for example, considering a threshold value of $th = 0.9$, one or multiple elements can be selected in the estimation of the quantization matrix. Another way to exploit this amount of knowledge is to help reducing the number of first quantization factors to be considered as candidates for the final estimation. Fixed a threshold th in the range $[0, 1]$, and denoted as p_i the output provided by the softmax function with respect to the event $q1 = i$, the smallest set of first quantization factors whose summation of related p_i is higher than th is selected. The quantization factors belonging to this set can be easily collected sorting probabilities p_i in decreasing order and computing the cumulative sum. For instance, considering the softmax outputs depicted in Fig. 2.26 and $th = 0.95$ only two $q1$ (1 and 5) are selected (see Fig. 2.27).

To better justify the design choices related to the regularization approach several tests were performed. Specifically, four double compressed datasets obtained cropping central patches with $d \in \{64, 128, 256, 512\}$, from 1000 images selected from BOSSBASE collection [69] were built. This collection was considered in the parameter setting to limit the overfitting with respect to the dataset employed to train the CNNs (i.e., RAISE [68]). To cope with real scenarios, double compression is performed employing custom tables from [73] by considering only the matrices with $q_{1i} \leq q_{1max} = 22$ and $i \in 1, 2, \ldots, 15$. Two different C_{data} terms were considered:

$$C_{data1} = 1 - \frac{1}{n} \sum_{j=i-\lfloor n/2 \rfloor}^{i+\lfloor n/2 \rfloor} p_j \tag{2.11}$$

$$C_{data2} = -\frac{1}{n} \sum_{j=i-\lfloor n/2 \rfloor}^{i+\lfloor n/2 \rfloor} \log p_j \tag{2.12}$$

where i indicates the position (zig-zag order) of the DCT term under analysis, n the number of considered neighbours and p_j the probability (i.e., softmax output) provided by the proposed CNN at position j related to q_{1j}.

Moreover, two different C_{reg} terms have been investigated:

$$C_{reg1} = \frac{1}{n-1} \sum_{j=i-\lfloor n/2 \rfloor+1}^{i+\lfloor n/2 \rfloor} |q1_j - q1_{j-1}| \tag{2.13}$$

$$C_{reg2} = \frac{2}{n-1} \sum_{j=i-\lfloor n/2 \rfloor+1}^{i+\lfloor n/2 \rfloor} \frac{|q1_j - q1_{j-1}|}{q1_j + q1_{j-1}} \tag{2.14}$$

where $q1_i$ is the first quantization candidate at position i (zig-zag order) under analysis.

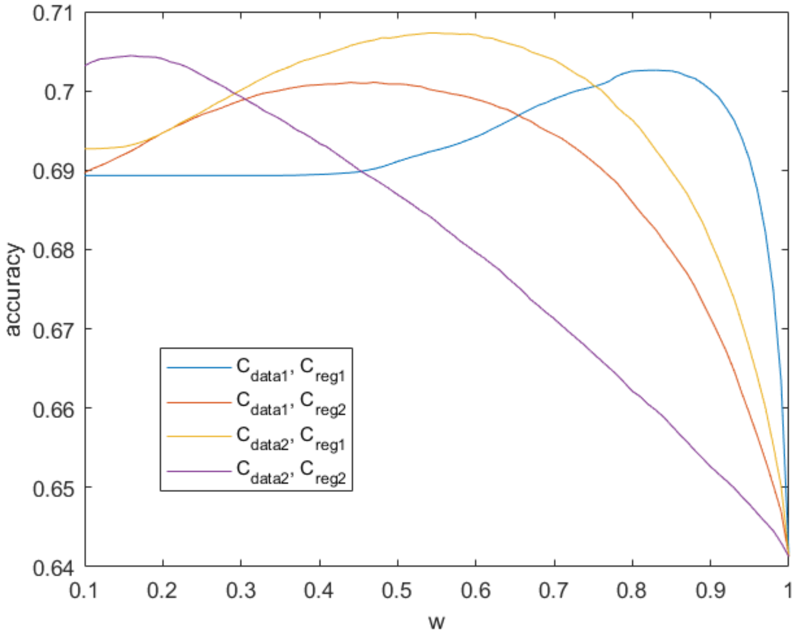


Figure 2.28: Average accuracy of the proposed regularization solution considering $n = 3$ neighbours and all the combinations of C_{data} and C_{reg} formulas.

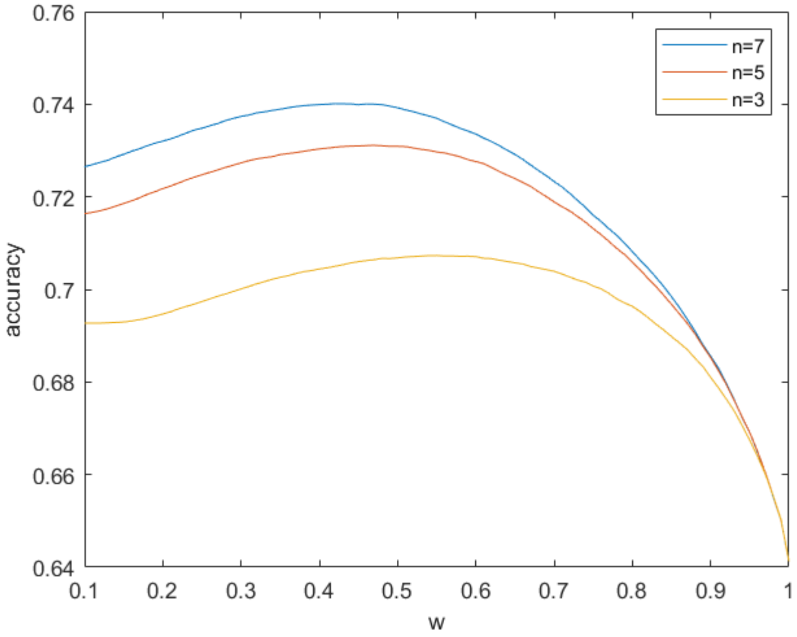


Figure 2.29: Average accuracy computed considering C_{data2} and C_{reg1} (best combinations) at varying n (i.e., number of neighbours). Note that $w = 1$ actually corresponds to results achieved without employing any regularization.

In Fig. 2.28 are reported the average accuracies obtained employing (2.10) considering all the possible combinations of C_{data} ((2.11), (2.12)) and C_{reg} ((2.13), (2.14)) with $n = 3$. For each weighting factor w, the average accuracy is computed taking into account the four aforementioned datasets ($d \in 64, 128, 256, 512$) and the DCT coefficients. Best performances are obtained with C_{data2} and C_{reg1} used for subsequent tests. An additional test was carried out with C_{data2} and C_{reg1} at varying the number of neighbours $n \in \{3, 5, 7\}$ and weighting factor w. Moreover, to make results comparable with respect to different number of neighbours n, only positions $i = 4, \ldots, 12$ are considered in the parameter setting tests. Note that, although the regularization strategy described considerably reduces the average number of combinations, worst case scenario has to be avoided. To this aim, the maximum number of allowed combinations per estimation was set to 10^6. As can be easily seen from Fig. 2.29, $n = 7$ and $w = 0.43$ provide the best results.

As already pointed out previously, the proposed solution differently than previous works in the field was specifically designed to work with a wide set of Q_2 matrices. For each $q2$, two CNNs related to DC and AC terms were trained by employing the parametric architecture depicted in Fig. 2.24. Although the designed CNN, taking as input DCT histograms, is not strictly limited to be used with a specific patch size, the accuracy with respect to different input parameters was also evaluated. More specifically, for each patch size $d \in \{64, 128, 256, 512\}$, $2 \cdot 22$ CNNs were trained with empirical histograms from double compressed JPEG images. Each dataset, one per patch size, was split into 80% training, 10% validation and 10% test and exploited to train 4 sets of $2 \cdot 22$ CNNs. Each group of CNNs is then trained with histograms obtained from input patches of the same size. It has to be noted that histograms containing no information [77] have been removed and were not considered in our tests. Results of the proposed CNNs at varying patch size and $\{q1, q2\}$ combinations are reported in Fig. 2.30 where average values are computed with respect to the first 15 DCT coefficients. The obtained accuracies strictly depend on the amount of information contained in the input histogram (higher at increasing patch size) and on the combination of $q1$ and $q2$ values (e.g., multiples). It is worth noting that the reported results were computed considering both training and test set related to input patches of the same size. To further study the performance of the proposed solution, additional tests were performed considering a scenario with a mismatch

CNN trained with Test set	64×64	128×128	256×256	512×512
64×64	**0.44**	0.43	0.41	0.38
128×128	0.53	**0.54**	0.53	0.50
256×256	0.57	0.62	**0.63**	0.62
512×512	0.58	0.65	0.70	**0.71**

Table 2.14: Accuracies of the CNNs trained with a specific patch size with respect to all the four generated test datasets ($d \in \{64, 128, 256, 512\}$). Average values were computed with respect to $q1 \in \{1, 2, \dots, 22\}$, $q2 \in \{1, 2, \dots, 22\}$ and the first 15 DCT terms. Each test set is a subset (10%) of the related one described before built employing constant matrices for first and second compressions and images from RAISE [68] collection.

CNN Test set	$ensemble_2$	$ensemble_4$	$ensemble_8$
64×64	**0.4447**	0.4443	0.4434
128×128	**0.5425**	0.5420	0.5411
256×256	**0.6302**	0.6295	0.6282
512×512	**0.7070**	0.7059	0.7046

Table 2.15: Accuracies of the CNN ensembles with respect to all the four generated test datasets ($d \in \{64, 128, 256, 512\}$). Average values were computed with respect to $q1 \in \{1, 2, \dots, 22\}$, $q2 \in \{1, 2, \dots, 22\}$ and the first 15 DCT terms.

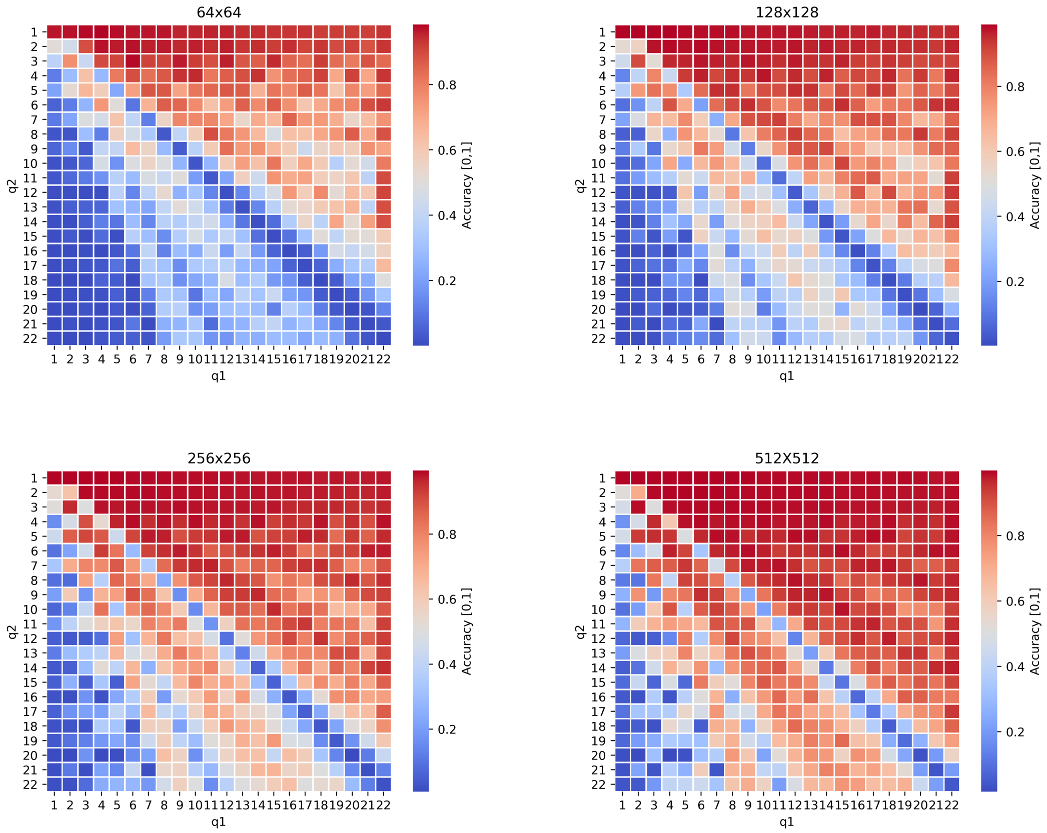


Figure 2.30: Accuracies of the trained CNNs at varying patch size and $q1$, $q2$ combinations. Average values were computed with respect to the first 15 DCT coefficients.

between train and test set patch size.

Table 2.14 shows the average accuracy, computed with respect to $q1 \in \{1, 2, \dots, 22\}$, $q2 \in \{1, 2, \dots, 22\}$ and the first 15 DCT terms, achieved by each couple dataset/CNN. As expected, it is evident that, for each dataset, the best result corresponds to the CNN trained with images of the same size. Note that scenarios involving both standard and custom quantization matrices actually select a subset of the possible $q1$, $q2$ combinations. Reported average accuracies are then not the same of the ones shown in Section 2.4.2.

As already pointed out in previously, the proposed solution, taking as input DCT histograms, does not strictly depend on a specific patch size. In order to improve the overall effectiveness and usability of the method, a single set of networks can be then trained with multiple patch sizes avoiding then the selection of a specific group of CNNs for each patch size. This improvement can be also really useful whenever input patch size actually differs from the one employed to train the models (e.g., 96×96, 384×512, etc.).

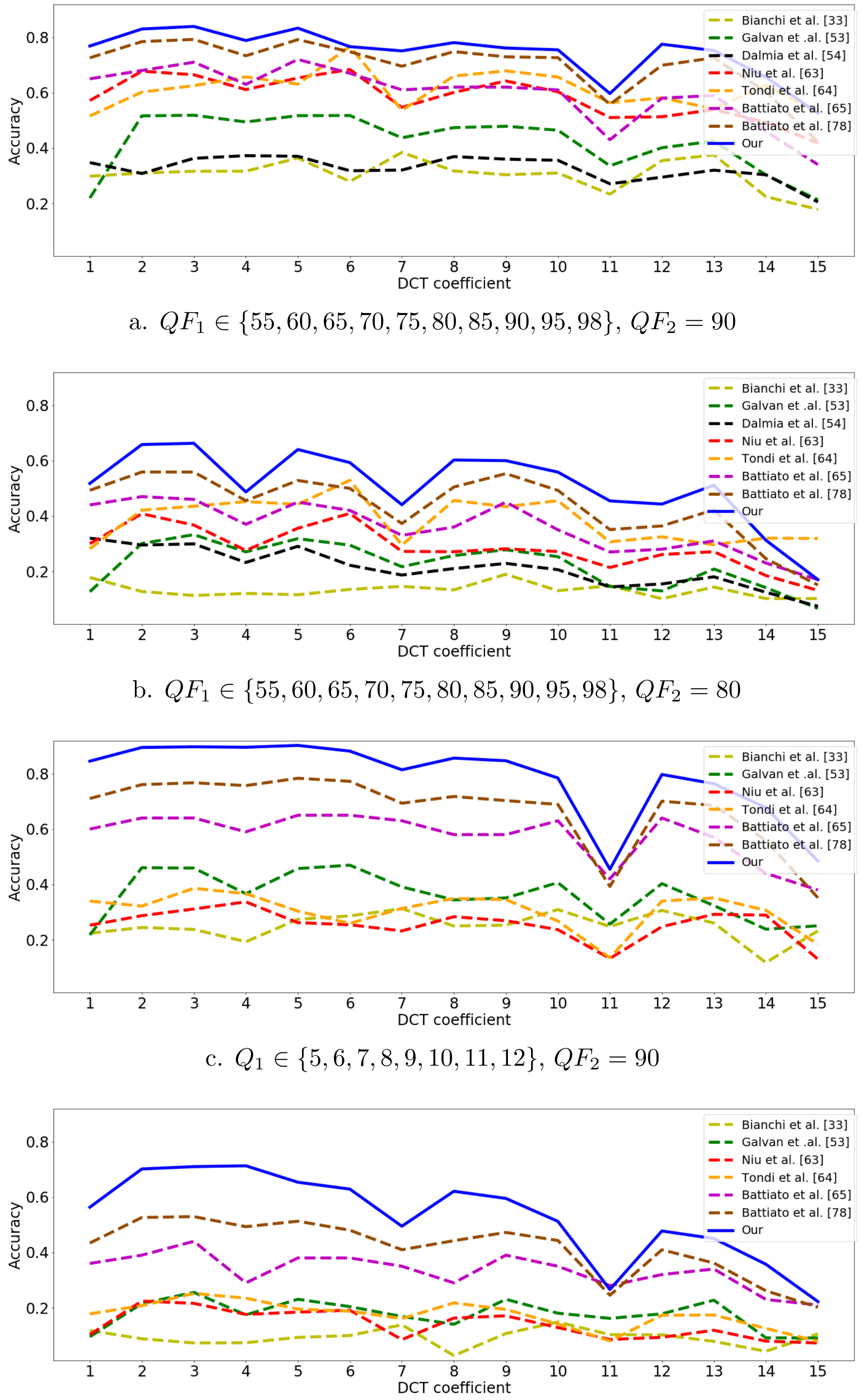
A novel dataset was then built by simply merging the collections employed before with $d \in \{64, 128, 256, 512\}$. However, the data to be handled requires a large amount of memory resources. In order to train the proposed method as the patch size varies, while exploiting all available data, a solution based on ensemble of CNNs was considered. Specifically, the merged dataset was split into 10 subdatasets, 8 employed for training, 1 for validation and 1 for test. Three CNN ensembles were considered with the following strategy: $ensemble_m$ represents an ensemble of m CNNs trained with $8/m$ training subdatasets $m \in \{2, 4, 8\}$. As reported in Tab. 2.15 all the proposed ensembles, differently than CNNs trained with fixed patch size, achieve satisfactory accuracy in all the considered test sets $d \in \{64, 128, 256, 512\}$. Although all the considered models achieve comparable accuracy ($ensemble_2$ slightly better than remaining ones), the solution with $m = 2$ does not increase considerably the execution time with respect to the networks trained with a fixed patch size.

Experimental Results

The effectiveness of the proposed method was demonstrated through a series of comparisons with state-of-the-art. All the experiments were run employing hardware equipped with a GPU NVIDIA TESLA K80. For comparisons we selected approaches belonging both statistical ([33, 53, 54, 65]) and machine learning based methods ([63], [64]); the public implementations of the method provided by the authors was employed to carry out the experiments. Moreover, to cope with real scenarios, both standard and custom matrices have been also considered in our tests. Tests described in the following were performed all with $k = 15$.

The aforementioned state-of-the-art solutions have been designed to work with specific patch sizes, then to properly compare the proposed method with them, a first series of tests was performed considering several scenarios involving 64×64 patches as input.

Specifically, 4 double compressed datasets were generated starting from random 64×64 patches cropped from RAISE collection [68] (one patch for each RAISE image):



a. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$

b. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$

c. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$

d. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

Figure 2.31: Accuracies of the same methods described in Table 2.16 and 2.17 at varying quantization factors $q1_i$ to be predicted. The values are averaged over all the QF_1/Q_1. Comparisons have been done with Bianchi et al. [33], Galvan et al. [53], Dalmia et al. [54], Niu et al. [63], Tondi et al. [64], Battiato et al. [65] (the method described in Section 2.3), and Battiato et al. [78] (the method described in Section 2.4.1). Dalmia et al. [54] is present only in comparisons a and due to the assumptions about the standard matrices of first compression in the provided implementation.

QF_1	$QF_2 = 90$								
	Our	Our reg.	*[78]*	*[33]*	*[53]*	*[54]*	*[63]*	*[64]*	*[65]*
55	**0.84**	0.80	0.77	0.53	0.52	0.45	0.00	0.00	0.62
60	0.81	**0.86**	0.82	0.53	0.56	0.47	0.64	0.53	0.66
65	**0.84**	**0.84**	0.81	0.54	0.57	0.49	0.54	0.81	0.68
70	0.83	**0.88**	0.85	0.43	0.57	0.51	0.66	0.66	0.70
75	0.89	0.89	0.85	0.41	0.63	0.53	0.77	**0.93**	0.75
80	**0.89**	0.87	0.83	0.29	0.61	0.45	0.81	0.67	0.75
85	0.86	**0.90**	0.85	0.14	0.74	0.36	0.81	0.88	0.72
90	**0.40**	0.12	0.24	0.00	0.00	0.00	0.02	0.02	0.23
95	0.50	0.62	0.52	0.11	0.00	0.00	**0.78**	0.68	0.40
98	0.36	0.76	0.57	0.00	0.00	0.00	0.76	**0.91**	0.41
MEAN	0.72	**0.75**	0.71	0.30	0.42	0.33	0.58	0.61	0.59
QF_1	$QF_2 = 80$								
55	**0.76**	0.61	0.58	0.36	0.37	0.37	0.24	0.52	0.49
60	**0.66**	0.65	0.60	0.27	0.37	0.38	0.50	0.31	0.47
65	0.67	**0.71**	0.65	0.19	0.41	0.43	0.31	0.38	0.52
70	0.72	**0.82**	0.75	0.19	0.50	0.49	0.50	0.70	0.66
75	**0.67**	0.65	0.56	0.07	0.56	0.45	0.15	0.58	0.47
80	**0.13**	0.05	0.11	0.00	0.00	0.00	0.00	0.04	0.04
85	**0.56**	0.53	0.34	0.19	0.00	0.00	0.04	0.14	0.31
90	0.34	0.40	0.19	0.06	0.00	0.00	**0.48**	0.37	0.21
95	0.11	0.35	0.30	0.00	0.00	0.00	**0.95**	0.37	0.19
98	0.09	0.43	0.42	0.01	0.00	0.00	0.21	**0.48**	0.19
MEAN	0.47	**0.52**	0.45	0.13	0.22	0.21	0.28	0.39	0.36

Table 2.16: Accuracies obtained by the proposed approach (with and without regularization) compared to Bianchi et al. [33], Galvan et al. [53], Dalmia et al. [54], Niu et al. [63], Tondi et al. [64], Battiato et al. [65] (the method described in Section 2.3) and Battiato et al. [78] (the method described in Section 2.4.1) with different combinations of QF_1/QF_2, considering standard quantization tables.

PS	$QF_2 = 90$							
	Our	Our reg.	[78]	[33]	[53]	[63]	[64]	[65]
5	**0.86**	0.84	0.78	0.56	0.58	0.05	0.10	0.69
6	**0.91**	0.88	0.82	0.46	0.60	0.07	0.13	0.75
7	**0.86**	0.85	0.83	0.41	0.58	0.07	0.09	0.70
8	**0.86**	**0.86**	0.81	0.25	0.65	0.10	0.13	0.74
9	0.67	**0.76**	0.61	0.02	0.47	0.02	0.05	0.47
10	0.60	**0.72**	0.50	0.19	0.00	0.25	0.33	0.43
11	0.40	0.69	0.52	0.04	0.00	0.69	**0.75**	0.40
12	0.36	0.76	0.57	0.04	0.00	0.75	**0.85**	0.41
MEAN	0.69	**0.80**	0.68	0.25	0.36	0.25	0.30	0.58
PS	$QF_2 = 80$							
5	**0.72**	**0.72**	0.68	0.26	0.46	0.07	0.08	0.57
6	0.53	**0.68**	0.54	0.05	0.41	0.02	0.06	0.42
7	0.72	**0.74**	0.68	0.15	0.48	0.08	0.10	0.56
8	0.33	**0.35**	0.22	0.03	0.03	0.01	0.04	0.18
9	0.49	**0.54**	0.28	0.19	0.00	0.07	0.13	0.28
10	0.23	**0.47**	0.20	0.00	0.00	0.40	0.16	0.17
11	0.09	**0.39**	0.38	0.01	0.00	0.24	0.38	0.23
12	0.09	0.43	0.42	0.01	0.00	0.21	**0.44**	0.26
MEAN	0.40	**0.54**	0.42	0.09	0.18	0.17	0.14	0.33

Table 2.17: Accuracies obtained by the proposed approach (with and without regularization) compared to Bianchi et al. [33], Galvan et al. [53], Niu et al. [63], Tondi et al. [64], Battiato et al. [65] and Battiato et al. [78] employing custom tables for first compression. The column PS refers to the quality (and then custom tables) used by Photoshop.

1. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$
2. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$
3. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$
4. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

where $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$ of 3) and 4) are referred to Photoshop's quantization matrices (version 20.0.4).

First and second dataset are related to the estimation of a classical scenario involving only standard quantization matrices in the first compression whereas, the last two employing also Photoshop's quantization tables, can be considered a more challenging test to verify the robustness of the considered methods with respect to real conditions. Dalmia et al. [54] has not been taken into account in the comparisons involving dataset 3) and 4) due to the assumptions about the standard matrices of first compression in the provided implementation.

Table 2.16 and Figures 2.31a, 2.31b demonstrate the robustness of the method in the scenario of standard quantization matrices, outperforming state-of-the-art solutions in almost all combinations. Same results are reported in Table 2.17 and Figures 2.31c, 2.31d which demonstrate an high degree of robustness with respect to custom quantization tables.

To further demonstrate the robustness of the proposed solution an additional test was performed. As reported in [73], analysing JPEG images downloaded from Internet, custom quantization tables are often employed in the compression pipeline. Specifically, Park et al. in [73] collected 1170 quantization matrices (100 standard and 1070 custom). This collection was exploited to carry out further tests: the matrices with $q1_i \leq q1_{max} = 22$ and $i \in \{1, 2, \ldots, 15\}$ were selected, sorted by the average of the first 15 quantization factors and then split into three sets of 291 elements (Low, Mid, High). Due to the double compression, 9 different combinations have been then considered. Moreover, to study the performance with respect to patch size and also employed dataset, 8 different input datasets were created: 4 different patch sizes (64×64, 128×128, 256×256, 512×512) cropped from RAISE [68] and UCID [72]. For each dataset, the quantization tables employed for double compression were randomly selected from the 291 available in the corresponding set (Low, Mid, High). It is worth to note that UCID dataset, due to the different

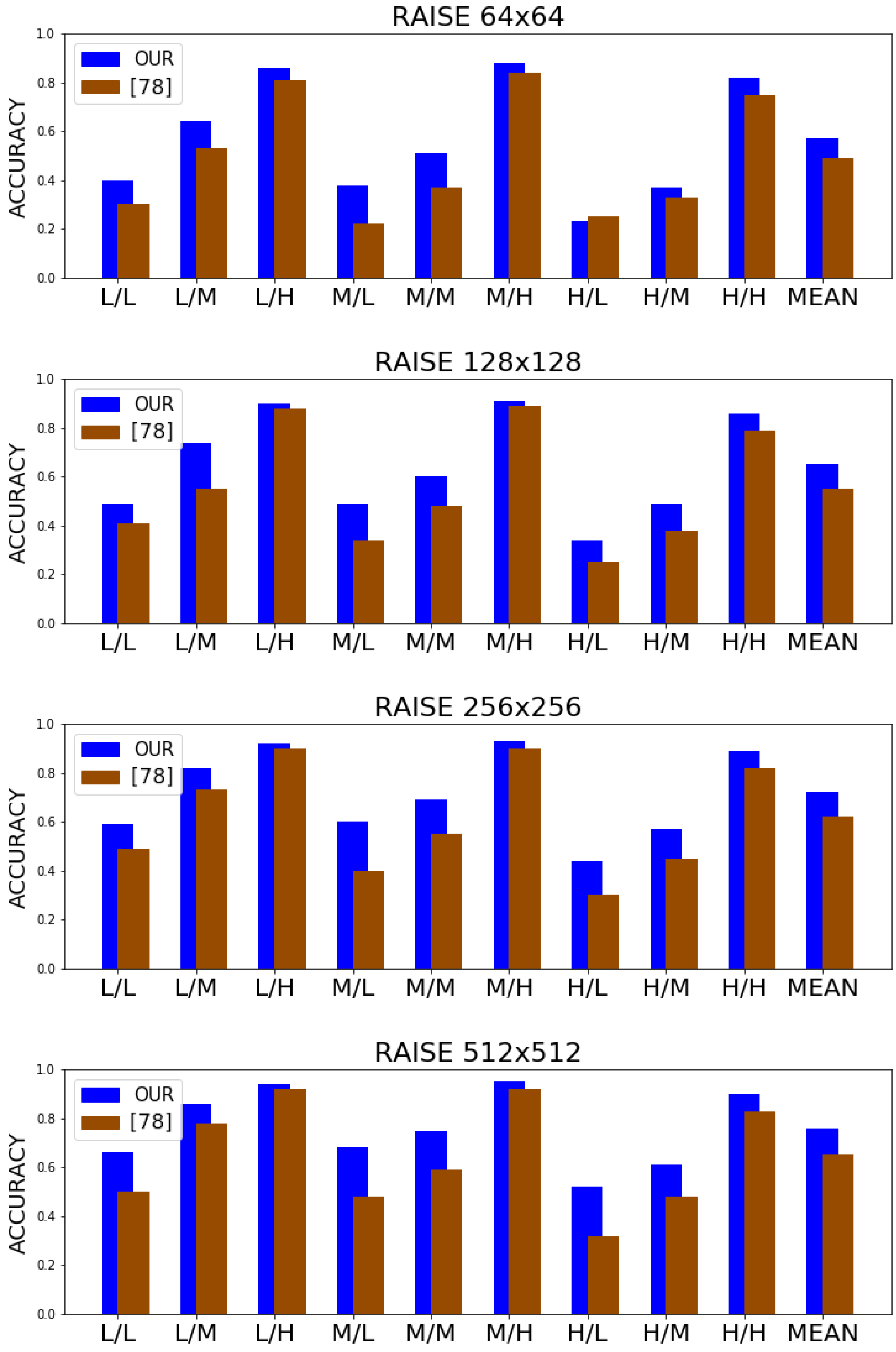


Figure 2.32: Comparison between the proposed solution and Battiato et al. [78] considering custom tables from Park et al. [73] and patches from RAISE dataset at varying $d \in \{64, 128, 256, 512\}$. L, M, and H represent respectively the sets of matrices Low, Mid and High described before.

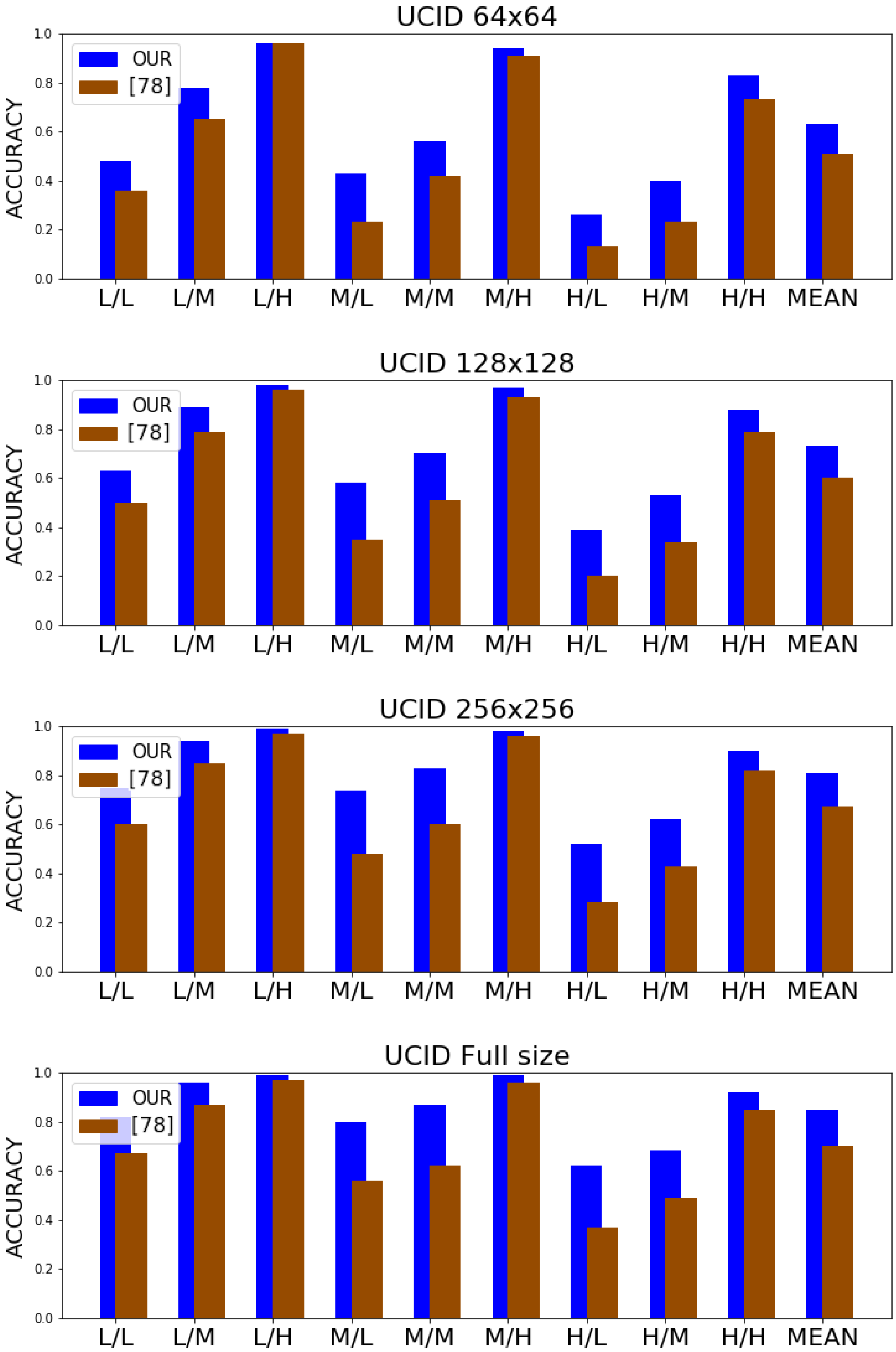


Figure 2.33: Comparison between the proposed solution and Battiato et al. [78] considering custom tables from Park et al. [73] and patches from UCID dataset at varying $d \in \{64, 128, 256, 512\}$. L, M, and H represent respectively the sets of matrices Low, Mid and High described before.

resolution of the original images used to extract patches vs. the collection employed to train the CNNs (i.e., RAISE [68]), allow us to verify the robustness of the proposed solution with respect to the variability of the dataset.

As can be seen from Figures 2.32 and 2.33 the proposed approach achieves satisfactory accuracy even in this challenging scenario. In addition, the results are closely related to the amount of information contained in the input histogram. A higher accuracy is therefore obtained as the patch size increases and with UCID dataset [72].

2.5 Summary

First Quantization Estimation (FQE) is an important task in forensic investigation activities. In this chapter several methods able to estimate the first quantization factors for JPEG double compressed images (in the aligned scenario) were described. To note that all methods work in the aligned scenario and does not consider the analysis of upload step related to the resizing of the image ([80]). Every method added a contribution to the state-of-the-art demonstrating the effectiveness of each one with the accuracy results. In particular, the last solution described (Section 2.4.2), represents an important improvement in terms of data organization. By proper collecting and training a neural architecture through 1D-histograms of DCT values (AC and DC terms) the proposed method exceeds the limits of classic neural network solutions. Furthermore, the regularization strategy shown in Section 2.3 before and Section 2.4.1 after, demonstrate how, in the aligned scenario, the method works better than other solutions for both custom and standard tables, becoming a viable solution to be employed in real-case scenarios. Finally the good results with different image sizes and the analysis conducted on their informative content suggested new points of view. Future works will be devoted to further extend the proposed solution to the non-aligned case and other forensics tasks such as tampering detection/localization.

Chapter 3

Document Identification

3.1 Introduction

Identification of printed materials is a critical and challenging issue from a security perspective, especially when dealing with documents such as banknotes, tickets or unique collector cards, which are suitable targets for ad hoc forgery. State-of-the-art methods demand high-cost, specialized industrial equipment, while a cost-efficient, fast and robust solution for document identification is a growing necessity in many contexts. The production process for common paper sheets involves the use of wood particles with successive application of other compounds. The inherently accidental imperfections produced, make the sheet almost unique and under some conditions, an authentic impression can be extracted. The massive demand in various contexts for robust identification methods [81, 82, 83, 84, 85, 86], makes fingerprint extraction from a sheet of paper an interesting and challenging research topic. Investigative scenarios in the forensic field [87, 88], could gain various benefits from the availability of such fingerprint.

In this chapter it will be presented a low-cost method for fingerprint extraction from a paper sheet. The method is composed by an acquisition framework, which defines the rules to digitize the paper support, and by a pipeline for fingerprint extraction, based on the Local Binary Pattern in the first version and on its variants in the extended version. The tests carried out, compared with the state-of-the-art demonstrate the goodness of the approach, also in presence of challenging scenario as the damage of the paper sheet.

3.2 State of the art

The identification of a document for legal purposes, could be done through the use of different techniques [87]. Although many of them are based on marks previously introduced in the analized material (e.g., watermarking) in the general scenario it is not present; for this reason the researcher moved their interest in techniques based on the fingerprints left on the surface of the paper without any specific embedding requirement. Most of the proposed techniques over the years require expensive industrial devices [89, 90]. The requirement to create a robust fingerprint extraction technique is the existence of low-cost physically unclonable functions (PUFs) which permit to obtain an intrinsic random physical feature. This feature, to be useful, must be able to return a unique response for the same request. The response must be unpredictable, even for an attacker with physical access to the object, by operating as a sort of random function. The paper surface presents an inherently unique structure, as it consists of overlapping and inter-twisted wood fibers. Hence, the imperfections of a paper sheet caused by the manufacturing process can be exploited to create a sheet's fingerprint. It has been proven that is extremely unlikely that two document surfaces created with the same raw materials will be identical, although they will present some similarities. This fingerprint makes forgery unfeasible, given that it is unique and virtually impossible to modify.

Buchanan et al. [91] proposed for the first time in 2005 the use of a fingerprinting technique for document identification; in their work the authors employed laser irradiation from four different angles and acquired the reflected energy to create a fingerprint. In [92] the authors proposed an improvement of [91] exploiting the correlation between the acquired energy signals. The Laser Surface Authentication (LSA) was introduced and employed for surface identification for the fist time by Cowburn in [93] and [94]. Clarkson et al. [95] extracted a feature vector from a collection of representative subsections and employing a Voronoi distribution to build the fingerprint. They do not extract a feature vector from a single region of the document, but they compute the feature vector from a collection of representative subsections, patches, of the document.

Samsul et al. [89] proposed a fingerprint extraction method, which exploits CCD sensors and laser speckle, to employ the visible pattern of bright and dark spots generated by interference of two or more light beams with different phases. A similar approach has been proposed by Sharma et al. [90]. In contrast to [89], they employed a microscope to acquire the speckle pattern. In recent years, CNN-based methods have achieved great performance in image recognition and classification, but have high complexity and require GPUs to perform training.

The aforementioned approaches work well for paper fingerprint extraction, but they require industrial and specific equipment. Recently, this limitation was overcome by the works of Toreini et al. [96] and Wong et al. [97]. In [97] the authors proposed a strategy to extract paper surface imperfections by exploiting multiple shots taken by a mobile camera under semi-controlled light conditions; subsequently, they investigated selected candidates through ad hoc mathematical models for each camera-captured image [98]. Unlike previous works, Toreini et al. [96] did not detect surface imperfections, but captured the random arrangement of the wood fibers within the paper sheet. To extract the paper pattern, they exploited a consumer camera and a backlit surface. However, they printed a bounding box on the analyzed paper to simplify the automatic texture registration. Since in real scenarios this registration strategy is not applicable, a different acquisition framework is needed. Based on the same filter of [96], Chen et al. in [99] exploited the microscopic features of wood fibers to obtain similar patterns, using expensive equipment based on double cameras. As already demonstrated in [100], the random disposition of wood fibers on paper sheets makes possible the construction of a fingerprint virtually impossible to tamper with; hence, given the limits of the previous works in terms of costs, acquisition constraints, and robustness, a novel fingerprint extraction strategy using specific low-cost image-acquisition equipment will be presented in next section.

3.3 Printed Document Identification

In this section a new method for document identification is presented; it has been published in both a preliminary [101] and extended version [102]. It proposes a fingerprint extraction method not requiring expensive industrial equipment (e.g., laser, microscopes), but only cheaper devices such as an RGB camera, as it is based on wood fiber translucent patterns. The method exploit the Local Binary Pattern

(LBP), a local descriptor computed by comparing a pixel, called pivot, to its neighbours in order to obtain a unique fingerprint. In the extendend version some variants of LBP were employed in order to achieve better results. In this chapter we referred to f-BP to indicate LBP and its variants. The main contributions of this method can be summarized as follows:

1. a new fingerprint extraction method, based on f-BP variants, which outperforms existing approaches in the field;
2. an optimization of a f-BP fingerprint that employs block subdivision and Principal Component Analysis (PCA);
3. a new public dataset that includes images acquired with both low-cost and high-end devices, showing wood fiber patterns, which is the only one available to the best of our knowledge;

3.3.1 Fingerprint Extraction Process

Lighting the surface to reveal the wood fibers is mandatory to successfully extract the pseudo-random pattern that is unique for every sheet. Yet, such patterns must be digitized and properly mathematically modeled to implement a robust document identification system, which is the goal of this method. Given a certain physical paper document d_i, the aim is to obtain a digital fingerprint F_i, namely a sequence of K ordered values $\{f_i^{(1)}, f_i^{(2)}, ..., f_i^{(K)}\}$, which is solely determined by correctly processing the digital image s_i, which is the acquisition of the document d_i. The overall proposed pipeline is summarized in Figure 3.1.

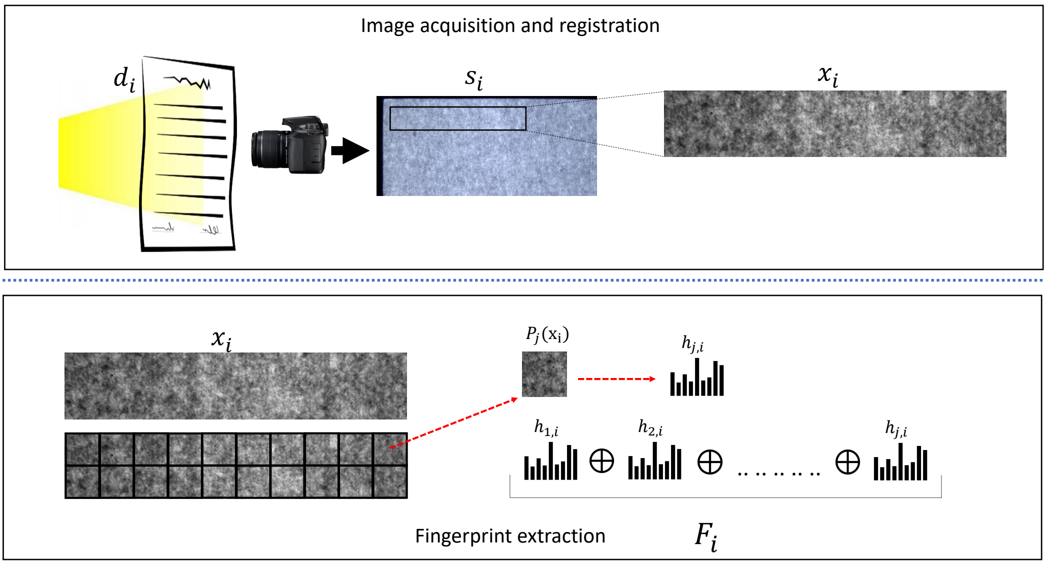


Figure 3.1: Overall pipeline of the proposed framework. The first row describes the process to acquire documents; the second row shows the fingerprint extraction process.

Document Digitization and Image Registration Considerations

The physical set of N documents $\mathfrak{D} = \{d_1, d_2, ...d_N\}$ was acquired using devices capable of capturing the wood fiber pattern by exploiting the translucent properties of the paper. Two different acquisition environments were employed to compare the performance of low-end and high-end equipment. Details about devices and related settings are provided in Section 3.3.2. The acquisition of a physical document d_i was performed in a semi-constrained environment; specifically, the documents must be roughly aligned with respect to the capturing device to ensure effective consequential registration. For readability reasons, the set of digitized versions of documents $\mathfrak{D} = \{d_1, d_2, ..., d_N\}$ can be defined as $\mathfrak{S} = \{s_1, s_2, ..., s_N\}$.

To successfully analyze the wood fiber pattern of a document d_i, the related digital image s_i must be registered. This step is critical as the paper fingerprint strongly depends on spatial information; hence, one must ensure that if a given document is acquired multiple times under the same setup, the system will process exactly the same region of the paper surface. To this aim, reference points were exploited (e.g., black bands in the acquired image) to rotate and properly crop s_i (see Section 3.3.1 for more details). After registration, a $W \times H$ sample from each document s_i was obtained, defined ad x_i, and the related set $X = \{x_1, x_2, ..., x_N\}$ was employed to build the fingerprint.

Extracting a Unique Fingerprint

The extraction of a unique fingerprint from a sample x_i is the process that encodes the texture information in a way that satisfies the following properties: (i) low complexity; (ii) encoding capabilities; (iii) robustness with respect to the missing parts. To this aim, the LBP descriptor and its variants [103] which have proven to fulfill all the aforementioned parameters, are employed,. These descriptors provide high capabilities in terms of discriminative power while maintaining low computational complexity and working almost perfectly even in the presence of slight variations on textures. In particular, LBP is a local descriptor that compares a pixel, called a pivot, to its n neighbors along the circle defined by a certain radius r [104]. The use of LBP for texture classification has grown in recent years, and a large set of LBP variants has been proposed [103]. Thus, the so-called f-BP variant aims to improve the accuracy and robustness for a specific task. The well-known local property makes the f-BP a flexible descriptor even in the presence of small perturbations, which is the fundamental requirement of the fingerprint we are looking for. Regardless of the f-BP, after pattern extraction the final descriptor is obtained by counting the times each pattern recurs, i.e., computing a histogram.

Histograms are compact and effective descriptors for a various number of tasks; however, they completely discard spatial information. To deal with this problem, x_i is first divided in M non-overlapping $p \times p$ patches and the histogram is computed separately for each patch $P_j(x_i)$ with $j = \{1, 2, ..., M\}$; hence the histogram $h_{i,j}$ represents the histogram of the j-th patch of the sample x_i. The relevance of spatiality is easily understood: if the document has certain types of defects (e.g., missing parts, tears, holes, noise), it is important that they do not affect the entire print, but only a portion. Therefore, the choice of the patch size p and the hyperparameters θ_f of the employed f-BP variant (e.g., the number of neighbors n and the radius r) have performance consequences. The size T of the histogram depends on the number of possible patterns the f-BP variant led. For example, employing classical LBP with $n = 8$ and $r = 1$ the number of possible patterns, and the histogram size T, is 256. Regarding the patch size p, large patches decrease the spatial information while small patches make the BP overly local and increase the complexity of the obtained fingerprint.

The final fingerprint F_i for document d_i can be obtained by concatenating all the histograms $h_{j,i}$ for $j = 1, 2, ..., M$:

$$F_i = \bigoplus_{j=1}^{M} h_{j,i} \tag{3.1}$$

The size K of F_i is $K = M \times T$, as M patches are obtained from M histograms of size T. The goal of this study is to test different f-BP variants and look for the parameters $\{W, H, p, \theta_f\}$ which lead to the most robust fingerprint.

3.3.2 Datasets for Document Identification and Fingerprint Testing

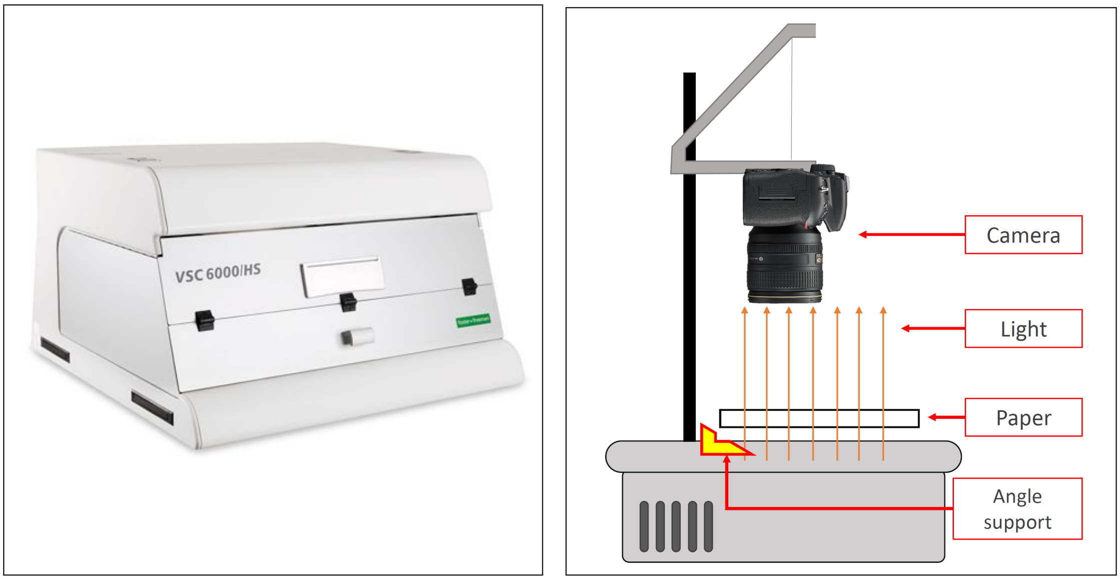
To evaluate the proposed approach and provide a great contribution to this research field, a new dataset consisting of 200 sheets of A4 paper arranged in groups of 40 and divided into 5 non-overlapping classes, is introduced. Each class is defined by two attributes: the manufacturer of the paper $b \in \{b_1, b_2, b_3, b_4\}$ and the weight or grammage (measured in g/m^2) $g \in \{80, 160, 200\}$. Thus, the obtained classes are the following: $(b_1, 80)$, $(b_2, 80)$, $(b_3, 80)$, $(b4, 160)$, $(b4, 200)$.

All the 200 documents in $\mathfrak{D}$ were then acquired multiple times using two different devices as described in Figure 3.2 and detailed in the next subsections.

Devices

To compare the performances achievable with high-end and low-end equipment, each document is digitized using two different devices. For the high-end case the Video Spectral Comparator 6000 (VSC) was employed while for the low-end one we used the Backlight Imaging Tool (BIT): an inexpensive overhead projector combined with a digital camera that we have meticulously designed.

The VSC consists of a main unit (Figure 3.2a) connected to a standard workstation. It provides several features and a number of different light sources to highlight paper details not normally visible under standard conditions. Table 3.1 shows VSC acquisition settings.



(a) Video Spectral Comparator 6000

(b) Backlight Imaging Tool

Figure 3.2: Devices employed for acquisitions.

Light	**Longpass**	**Mag**	**Exposure**	**Brightness**
Transmitted	VIS	2.18	Auto	60

Table 3.1: VSC Settings

The BIT consists of an overhead projector that is the lighting source and an RGB camera hanging from the projector arm. The employed camera is a Nikon D3300 equipped with a Nikon DX VR 15 mm–55 mm 1: 3.5–5.6 GII lens. Settings details are listed in Table 3.2.

Dataset Acquisition

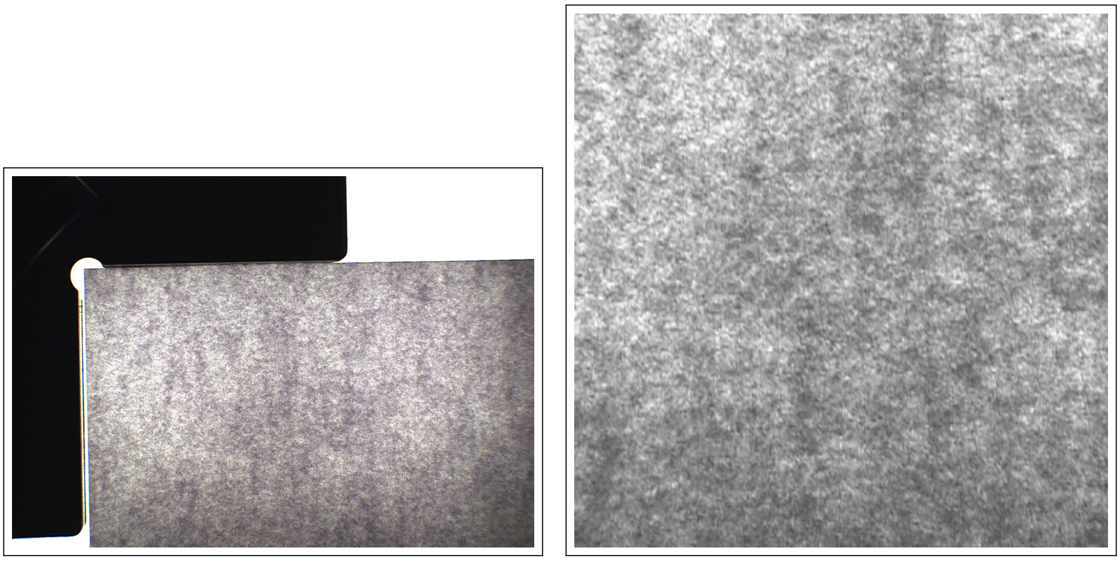
For the sake of clarity, the terms $\mathfrak{S}_{VSC}$ and $\mathfrak{S}_{BIT}$ will be employed for referring to the digital acquisitions made by the VSC and the BIT respectively. The overall acquisition pipeline of the dataset is depicted in Figure 3.1. As expected, $\mathfrak{S}_{VSC}$ and $\mathfrak{S}_{BIT}$ show different contrast and sharpness.

$\mathfrak{S}_{VSC}$ consists of 200 documents acquired twice, for a total of 400 acquisitions (Table 3.3). The result of a single acquisition is a bitmap image of 1292 × 978 pixels and 300 dot per inch (dpi), as reported in Figure 3.3a. $\mathfrak{S}_{BIT}$ consists of 200 documents acquired 8 times. However, under-powered light in the BIT does

Acquisition	Exp. Time	Opening/ISO/VR	Exp. Compensation	White Balance
RAW + JPEG	1/25	F29/100/ON	−5.0	Incandescence

Table 3.2: BIT Settings

not allow the extraction of the translucent pattern from paper with grammage 160 or 200. Thus, only the 120 documents with grammage 80 were considered for a total of 960 acquisitions with a resolution of 6000 × 4000 pixels and 300 dpi (Table 3.3). Figure 3.4a shows a raw acquisition, where the black bands, used for image registration, are visible.



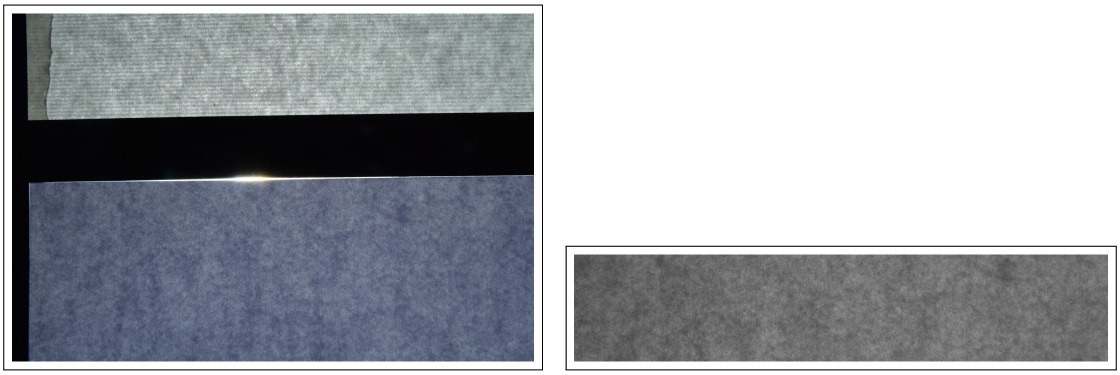
(a) Before registration

(b) After registration

Figure 3.3: Document acquisition with VSC before registration (**a**) and after registration (**b**).

Image Registration

The acquisition of black bands on the outside of the paper surface was intentionally performed to discriminate selectively the pixels in the external area and to easily obtain a set of registered images. All the raw images in $\mathfrak{S}_{VSC}$ and $\mathfrak{S}_{BIT}$ were converted to grayscale. A luminance threshold is first used to find the upper left corner (y_0, y_1) of the paper. Secondly, the image anchored in position (y_0+u, y_1+u) is cropped, where u is the minimum offset to perform a cropping by excluding the outer area. The value of u is variable: the larger the external area acquired is, the greater its value will be. Images acquired by means of the VSC are cropped into



(a) Before registration

(b) After registration

Figure 3.4: Document acquisition with BIT before (**a**) and after registration (**b**).

patches of 400 × 400, while the ones acquired with the BIT are cropped into patches of 5000 × 1000 pixels. Ultimately, we obtained X_{VSC}, the set of 400 registered samples from VSC, and X_{BIT}, the set of 960 registered samples from BIT. Source examples are shown in Figures 3.3b and 3.4b.

	VSC	**BIT**
CLASS	**ACQUISITIONS**	**ACQUISITIONS**
{e,80}	2	8
{f,80}	2	8
{u,80}	2	8
{m,160}	2	-
{m,200}	2	-
DEVICE IMG	$10 \times 40 = 400$	$24 \times 40 = 960$

Table 3.3: Dataset Table.

3.3.3 Experimental results

In order to thoroughly evaluate the proposed fingerprint extraction approach, the analysis of the datasets described in 3.3.2 were performed in terms of recognition tests. Since each document was acquired multiple times (i.e., twice for the VSC and 8 times for the BIT), a fingerprint reference dataset was built to deal with the recognition task; such reference datasets consist of only one sample per document while the rest of the samples were used for querying it. A given document d with extracted fingerprint F_a will have a correct match with the closest element in the reference dataset F_b, if both F_a and F_b "belong" to the document d; in other words, a correct match occurs if s_a and s_b are different acquisitions of the same document. The recognition test performances are measured using the well-known accuracy metric defined as the rate of queries that obtain a correct match. The adopted similarity measure for fingerprints was the Bhattacharyya distance [105], which is typically and effectively employed for problems where probability distributions must be compared. However, to better assess the effectiveness of the proposed fingerprint, four different recognition experiments are performed as detailed in the following. First, the original LBP was employed to compare the recognition accuracy on both datasets (VSC and BIT) obtaining the demonstration of device invariance. Given this result, a comparison was performed only on the BIT dataset employing LBP fingerprints computed as in [101] vs. the three other LBP variants, i.e., LTP [106], SBP [107] and CLBP [108]. Moreover, also the fingerprint robustness was investigated. To this aim, a challenging scenario was created in which query samples were intentionally altered by removing some pixels from the digital image to simulate physical damage of the paper (e.g., tears, holes). An optimization in terms of footprint size was also performed and tested by leveraging principal component analysis (PCA) [109].

Dataset Comparison

The validity of the LBP-based fingerprint extraction method has been demonstrated starting from the work of Guarnera et al. [101], which is representative of the state of the art. Table 3.4 shows the overall accuracy obtained in the recognition tests performed on both datasets: 96.5% and 99.2% for VSC and BIT, respectively. Although samples from different datasets have different patch sizes, the best for both datasets was 100×100. This is a reasonable trade-off to preserve local spatial information. The accuracy on the BIT dataset is slightly higher than the accuracy obtained on the VSC. This demonstrates that the robustness of the fingerprint does not depend on the acquisition settings nor on the device.

Comparisons among LBP Variants

As introduced earlier, many variations of LBP have been proposed for texture analysis. Among them, LTP [106], SBP [107] and CLBP [108] were selected for the

Dataset	Patch Size	Number of Neighbors	Radius	LBP Type	Accuracy
VSC	100	32	42	uniform	96.5%
BIT	100	24	12	default	99.2%

Table 3.4: Best configuration parameters and accuracy of recognition test in VSC and BIT datasets.

experiments described below. In the previous section, the independence of the proposed fingerprint from the acquisition device was demonstrated. Based on this evidence, in future experiments, only the BIT dataset will be used, due to the larger number of available samples .

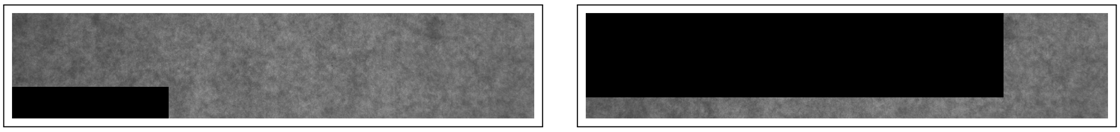
The results in terms of accuracy are reported in Table 3.5 where CLBP and SBP show an improvement in terms of performance vs. LBP, by achieving an accuracy of 99.7% and 99.4%, respectively. It is worth noting that LBP is the employed method of [101] to extract the fingerprint, so the aforementioned results represent the over-performance with respect to the state of the art. The LTP tends to perform better than LBP when the texture presents regions that are uniform (i.e., low variance), as described in the literature. It is worth noting that wood fiber models exhibit a high variance, which explains the weaker results for this descriptor. As expected, SBP, which is a generalization of the common binary pattern, obtains accuracy results of (99.4%) which are somewhat better than LBP. At last, the best performance was obtained by CLBP (99.7%) even if it delivers the largest fingerprint in terms of histogram dimensions (number of bins).

	LTP	CLBP	SBP
Accuracy	90.83%	99.7%	99.4%

Table 3.5: Recognition test accuracy of the test carried out in BIT dataset employing LTP, SBP and CLBP.

Tests on Noisy Environment

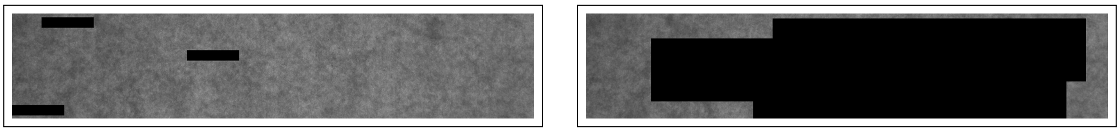
The proposed method for fingerprint extraction has been tested under controlled conditions to correctly assess what was expected to occur in real cases, i.e. when a document has undergone some degree of alteration between the first and subsequent fingerprint extraction. Hence, the original fingerprint of the document may be very different from the latter. There were two types of damage simulated on the paper for this purpose: tears and stain. The “tear” simulates a loss of information starting at one corner of a x_i sheet sample by replacing this loss with black pixels, while the “stain” introduces random black blocks on the sample to simulate holes or stains. For both, the so-called degree represents the size of black area: the maximum degree corresponds to about 75% of the full sample to be removed (see Figures 3.5 and 3.6). Given the aforementioned alterations, a new recognition test on the BIT dataset was carried out, which includes 120 samples without any alterations on the fingerprint database and other 960 samples with alterations that were used to query the database. The results are reported in Figures 3.7 and 3.8 further proving the robustness of the proposed fingerprint, particularly the CLBP-based fingerprint that once again achieves the best performance.



(a) Degree 1.

(b) Degree 11.

Figure 3.5: Examples of altered documents with simulations of tear damage; in particular (**a**) represents the first degree of damage while (**b**) the last (e.g., 11).



(a) Degree 1.

(b) Degree 11.

Figure 3.6: Examples of altered documents with simulations of stain damage; in particular (**a**) represents the first degree of damage while (**b**) the last (e.g., 11).

Fingerprint Dimensions Optimization

For all the tests outlined in the previous sections, the pipeline described in Figure 3.1 has been used with the following settings: images were cropped into patches of 100 × 100 pixels; number of neighbors for CLBP were $n = 12$ and radius was $r = 6$. These settings increased the number of patches in the BIT dataset to 500 thus, a histogram of 8194 elements was computed for each patch. This results in a fingerprint with a size of $500 \times 8194 = 4.097.000$ elements whose storage occupancy

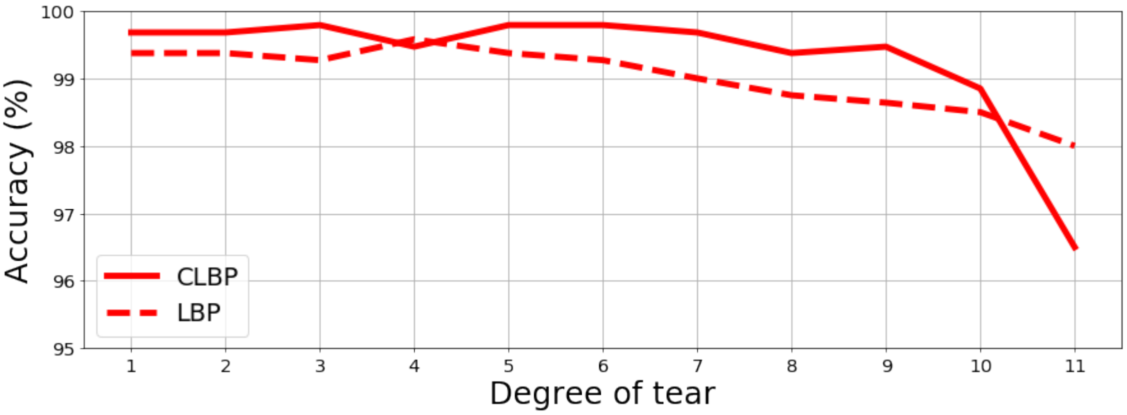


Figure 3.7: Accuracy employing CLB and LBP VS degrees of tear alteration.

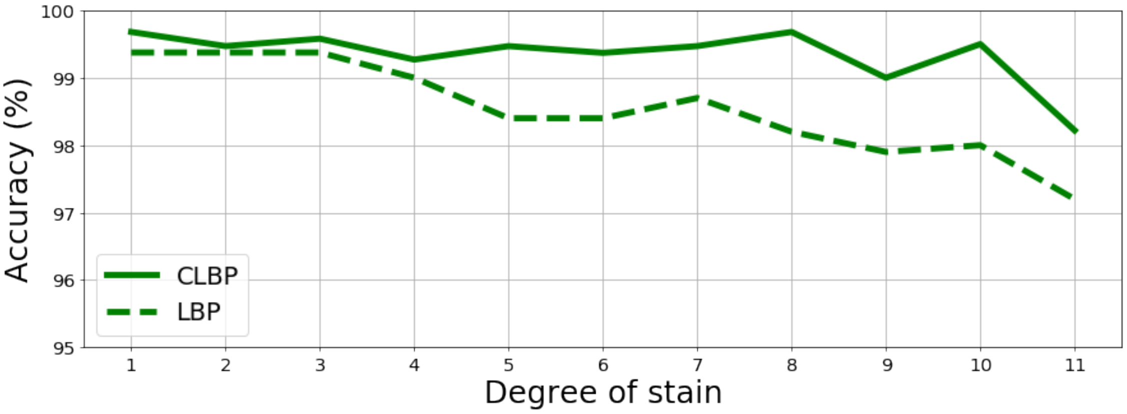


Figure 3.8: Accuracy employing CLB and LBP VS degrees of stain alteration.

is about 8.3 MB. Since the fingerprint with the proposed method could be even larger depending on parameters and since large fingerprints decrease efficiency, some optimization strategies to reduce it have been investigated.

The simplest strategy to reduce the fingerprint size was to increase the patch size p; however, this could not guarantee the same accuracy performance. Table 3.6 shows the results obtained using values greater than p while controlling the fingerprint size. The outcome analysis showed that the setting with $p = 200$ exhibit a similar performance to that of $p = 100$ (i.e., only a 0.4% of decrease in accuracy) by reducing the size by 25%, from 4.097.000 to 1.024.250 elements, that can be stored in 2.2 MB. However, as mentioned in previous sections, using larger patches does not preserve the spatial information and actually shows a huge drop in accuracy (e.g., 67.6% for $p = 500$).

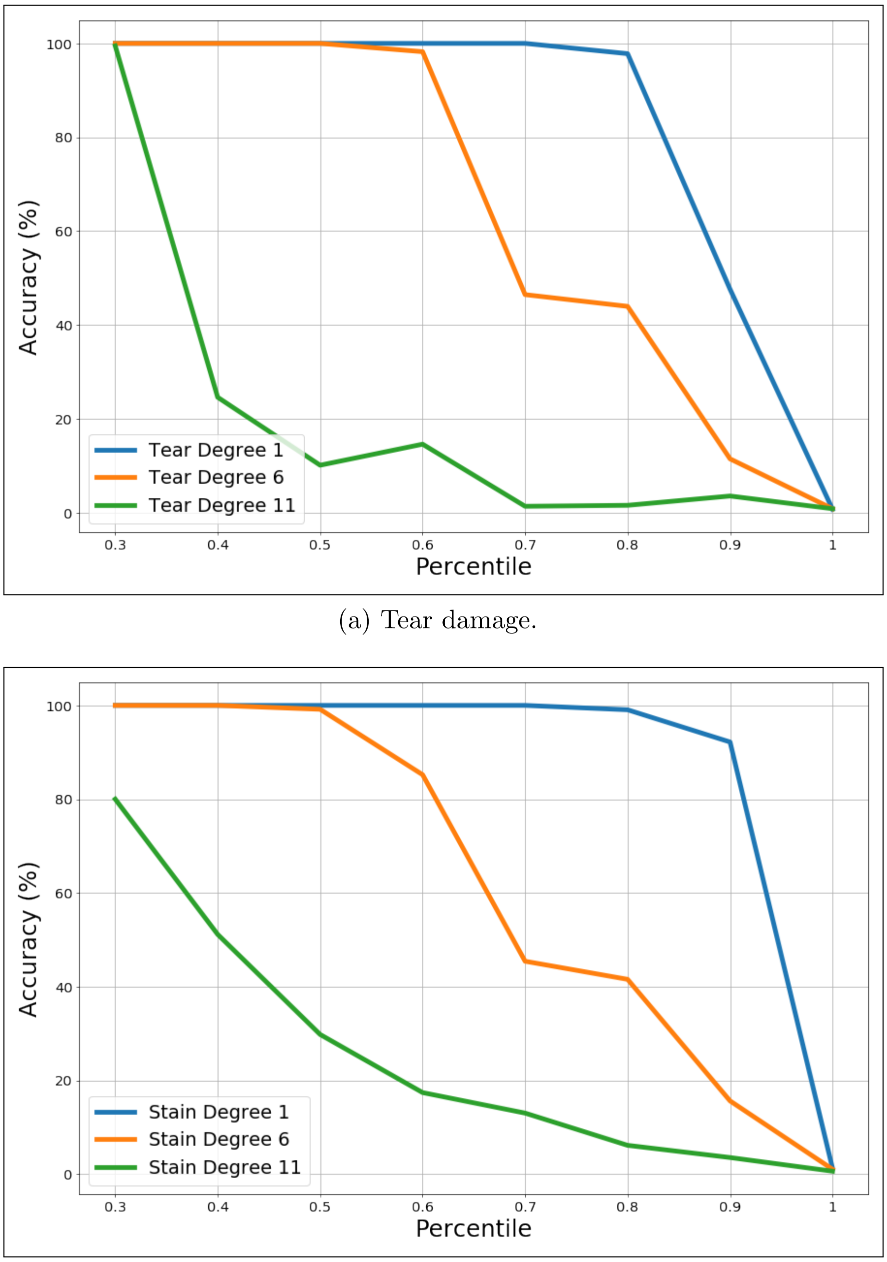
Patch Size	**Number of Bin**	**Accuracy**	**Storage Occupancy (MB)**
100	4.097.000	99.7%	8
200	1.024.250	99.3%	2.2
250	655.520	97.9%	1.5
500	163.880	67.6%	0.4

Table 3.6: Accuracy and size of CLBP fingerprints to vary of patch size.

In order to optimize the size of the fingerprint while avoiding a large loss in accuracy, we employed principal component analysis (PCA). As we know, PCA reduces the size by projecting each data point onto only a few of the principal components to obtain smaller data thus preserving most of the variance. In short, it reduces the size by preserving most of the information, which better explains a certain phenomenon. PCA is applied to each histogram $h_{j,i}$ previously obtained using CLPB. Hence, such histograms are drastically reduced in terms of dimensions. First, for testing purposes, all the 120 samples included in the fingerprint database are used to fit the PCA model. By employing the well-known explained variance analysis, we found that 95% of the information can be preserved using the first 32 principal components (also known as features), despite the original 8194. However, PCA shift the histograms into a geometric space where the Bhattacharyya distance becomes less efficient. To handle this problem, the recognition test was performed by means of the Euclidean distance. To verify the quality of reduction, the same recognition tests, as described in the previous sections, were carried out with the now-reduced fingerprints, delivering an accuracy of 97.97% with only 16,000 elements while maintaining the excellent performance of the not-reduced fingerprints case. It is worth noting that the PCA model was built using all the samples of each of the 120 documents in BIT. This could generate a PCA model overfitted on the data. Thus, a further test was performed using only the 50% of the dataset (60 documents) to fit the PCA model, and we queried the reference dataset with the samples which come from the remaining 50%. In this case, it was found that 95% of information can be preserved using the first 40 principal components for each patch. Recognition tests confirmed the results obtained with the PCA model built on all the 120 documents (97.97% of accuracy). It is important to note that although in the fingerprint comparison we also consider the missing parts when an alteration occurs, this does not heavily affect the Bhattacharyya distance between two fingerprints. On the contrary, the Euclidean distance is affected by this. In fact, the Euclidean distance calculated between an unaltered fingerprint of a document and an altered fingerprint of the same document exhibits extremely higher values, which impacts on the accuracy performance. To overcome this latter problem, a custom Euclidean distance was employed, where only a part of the fingerprint elements is considered in distance computation. Specifically, the differences between each element of the two fingerprints is computed and, subsequently, we sorted those differences by considering only a certain percentage of the lower ones. This percentage depends on the dimensions of the missing part, but this information is known by the operator during the identification phase, because in a real document the altered parts are visible. Figures 3.9 a,b report the accuracy (vertical axis) when varying the percentage of elements included in distance computation (horizontal axis). The obtained results also suggest how to maintain a high accuracy according to the alteration degree. For example, in an average scenario of damage (orange lines) the 50% of distance is needed to maintain the accuracy over the 99%.

3.3.4 Fingerprint Robustness Analysis

The carried-out recognition tests started from the assumption that every query fingerprint F_q could find a corresponding one F_x from the same document in the database of previously extracted fingerprints. A real-world scenario might present some differences: the query fingerprint F_q could not find a correspondent F_x and



(a) Tear damage.

(b) Stain damage.

Figure 3.9: Accuracy variability of different percentiles on tear (**a**) and stain (**b**) damages.

then the closest is meaningless (it is the closest but it is a fingerprint extracted from another document). Hence, additional information is needed; given the distance $\bar{e}$ between two samples. To solve this problem, $\bar{e}$ was analyzed in all the experiments presented above; in particular, starting from the fingerprints extracted by the images acquired with BIT device (e.g., 960), the distances obtained in the simulated damage-free tests using CLBP and LBP were analyzed, considering three types of distances:

- $\bar{e}_0$: distance obtained between F_q and F_x, both extracted from the same document, when F_x is the closest fingerprint in the recognition test.
- $\bar{e}_1$: distance obtained between F_q and F_x, both extracted from the same document, when F_x is not the closest fingerprint in the recognition test.
- $\bar{e}_{null}$: distance obtained between F_q and F_x, extracted from different documents.

Given 840 different F_q, 120 distances have been computed for each of them. For every F_q analyzed, two results were obtained:

- the closest fingerprint F_x is extracted from the same document of F_q, and then the distance between them is classified as $\bar{e}_0$ and the others 119 distances are classified as $\bar{e}_{null}$.
- the closest fingerprint F_x is not extracted from the same document of F_q, and then the distance between them is classified as $\bar{e}_1$ and the others 119 distances are classified as $\bar{e}_{null}$.

It is easy to figure out that the population of $\bar{e}_{null}$ is much bigger than $\bar{e}_0$ and $\bar{e}_1$, whose sum is exactly 840. Figures 3.10 a,b represent the plot of distances $\bar{e}_0$, $\bar{e}_1$, $\bar{e}_{null}$ in both tests (LBP and CLBP). The graphs were cut because the populations are unbalanced and because the focus of the analysis is on the intersections of the two curves. In these graphs, it is possible to detect two nearly entirely separated Gaussians. The intersection between them (the tail of the green Gaussian, bounded by the orange and blue lines) represents an area of uncertainty. It is worth noting that the position of $\bar{e}_1$ in both cases (LBP and CLBP) is within this area which confirms the meaning of the distance: the smaller the distance with the closest print, the greater the chance that the prints are extracted from the same document. The concept of low/big depends of course on the descriptor used; in the forensic domain it is important to measure the degree of uncertainty whenever it is available. The percentage of uncertainty zone z and the percentage r of $\bar{e}_0$ inside it gives a further degree of confidence and it is variable for each descriptor. Given a descriptor the pair (z,r) can be employed to describe its robustness. CLBP has the $\bar{e}_0$ range between 0.286 and 0.338 and uncertainty zone between 0.331 and 0.338 and then $z = 13.46\%$, while $r = 2.62\%$ due to 22 $\bar{e}_0$ inside uncertainty zone on 837 total; LBP has $z = 13.56\%$ and $r = 4.92\%$. Table 3.7 shows the analysis for each binary pattern tested.

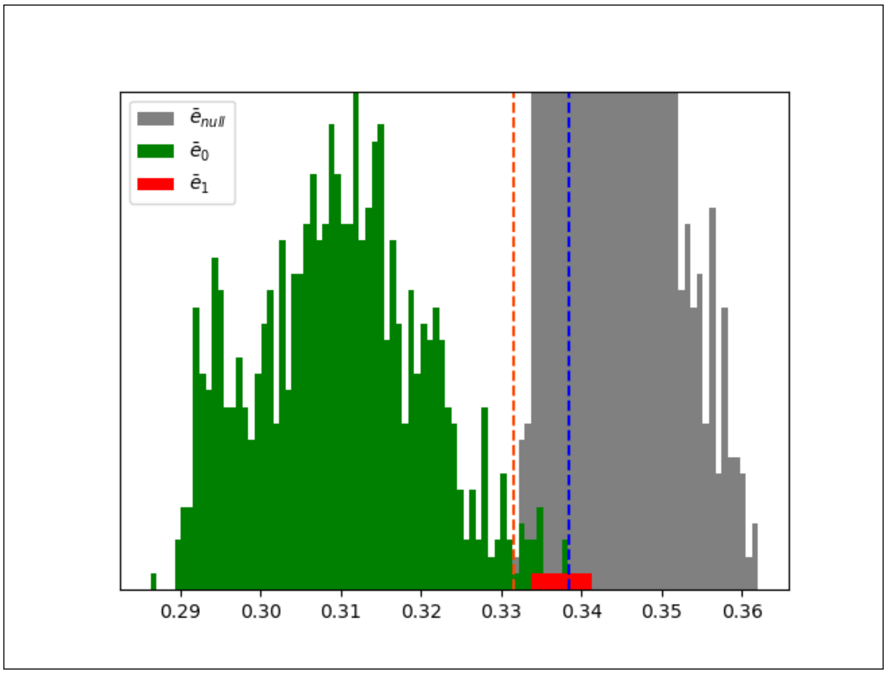
Descriptor	**z**	**r**
LBP	13.93 %	4.91 %
CLBP	13.32 %	2.62 %
SBP	15.72 %	3.13 %
LTP	91.38 %	91.05 %

Table 3.7: Percentage of uncertainty zone (z) and percentage of $\bar{e}_0$ inside it (r) for each analyzed descriptor.

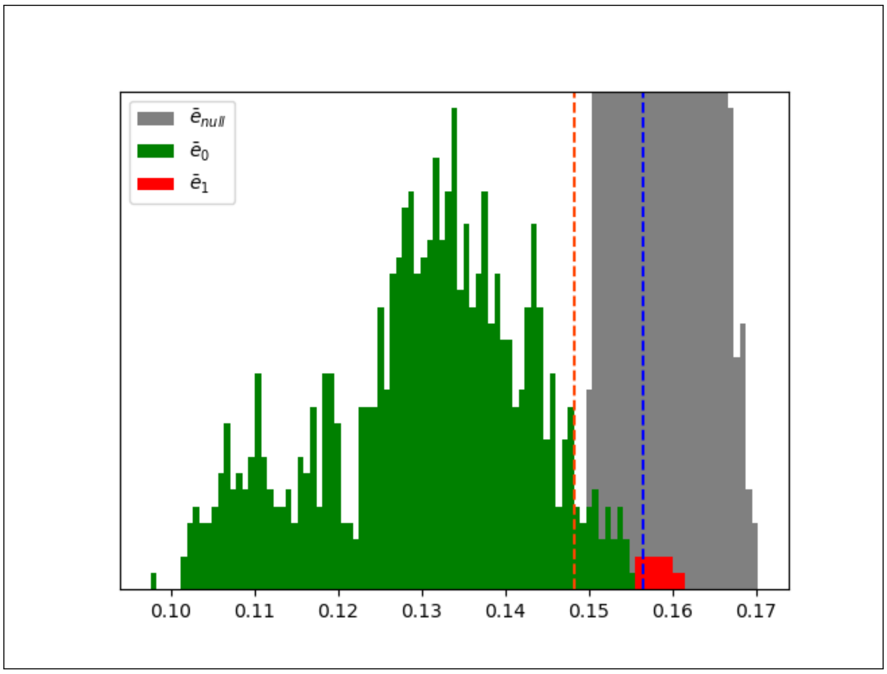
In addition, a cross-dataset analysis has been performed to determine if there was any correlation between input and descriptor efficiency. The textures with the distance within z have been analyzed: CLBP has 22 distance out of 837 while LBP 41 out of 835. 13 are shared whereas others are near to z meaning that bad texture (in terms of acquisition) will have a bad distances (close or within z), regardless of the descriptor.

3.4 Summary

In this chapter, a novel approach for document identification was proposed. The method employs variants of binary pattern descriptors (e.g., LBP, LTP, SBP, CLBP) to obtain an appropriate fingerprint to uniquely recognize the input document, but at the same time, be easily manageable. For this reason, an additional analysis



(a) Tear damage.



(b) Stain damage.

Figure 3.10: Accuracy variability of different percentiles on tear (**a**) and stain (**b**) damages. For both the plots x-axis represents the values of the distances obtained and y-axis the number of occurrences. $\bar{e}_{null}$, $\bar{e}_0$ and $\bar{e}_1$ are represented by gray, green and red respectively.

was conducted to optimize the fingerprint in terms of dimensions; it was based on PCA which has confirmed almost the same accuracy, reducing the fingerprint size to less than 1/100 of the original. To demonstrate the robustness of the method, the dataset was expanded by including more noisy samples, demonstrating the value of the proposed technique in real case scenarios and better results with respect to the state of the art. At last, further analysis on the significance of distances was conducted to generalize the recognition test.

Chapter 4

Conclusion

The widespread of Social Networks and Instant Messaging Platforms permitted the sharing of a huge number of media files and in particular images. At the same time, the usability of digital media editing tools have increased the possibility of malicious attacks on the shared media files bringing misinformation and creating social problems in many cases. In this scenario the Forensic Science has grown developing solutions able to cope with some of the aforementioned problems, even if many of them are still open today. Moreover, some solutions are able to resolve a specific problem: if the problem evolves or produce an anti-forensics solution the previous solution is useless and therefore needs to be updated.

The discussion on the usage of the media inside the court is based on a main problem: the authenticity of the evidence. Every type of evidence needs to be authenticated to be employed in a process, but depending on the evidence, the meaning of it could change. In the case of an image, it could be defined authentic if the semantic meaning is equal with respect to the original (intact image). Hence when an image becomes an investigation object some considerations about it could be done: is it an intact image? If not, is it authentic? Is it possible to reconstruct the history from the acquisition until now?

In order to answer to the previous questions some techniques were developed. A first analysis involves metadata and compression type, and in particular JPEG compression. Today, JPEG is the most widespread compression employed, and the usage of it by the Social Network Platforms has increased its importance. Most of the forensic analysis in fact, are aimed at this type of images.

First Quantization Estimation is one of the most faced task in JPEG analysis; its first objective is to identify the camera model in order to make assumptions about the person who took the photo but, as demonstrated over the years FQE, is useful often for tampering detection. FQE methods could be splitted in two main classes: the methods based on statistical analysis and the more recent methods based on Machine Learning. Both classes suffer of specific issues: the first does not permit to overcome an accuracy limit due to the extracted feature type while the second suffer of overfitting.

The generation of new methods for FQE was the main goal of this PhD thesis. Preliminary studies were done on the JPEG pipeline, in order to understand what type of feature gives the most important information about the image's history. The Discrete Cosine Transform was deeply analyzed in the first year of PhD; its coefficients inside the image represent the feature with most presence of compression traces, which we exploited also for JPEG Double Quantization Detection ([101]).

By exploiting and analyzing the overall DCT coefficients statistics, it is possible to understand how they describe the transformations of JPEG compression but also the correlation between them and the image content. In our first FQE approach ([65]) we employed this insight to build a method which uses the content (from the input image) simulating opportunely the double compression; in this way, the DCT content is similar to the real and from the simulated double compressions the most similar is chosen to estimate the first quantization factor.

The recent widespread of Machine Learning has led researchers to exploit solutions based on it. The power of this type of approach is the possibility to obtain the right features based on the objective. Machine Learning based techniques for FQE overcame previously methods but only in specific conditions, such as definite quality factor or exact input dimension. These limits have been analized and resolved changing the input to train the CNN in last PhD works: we employed as training data the DCT coefficients distributions to overcome the size limit generating a double compression strategy to simulate a huge number of quantization matrices. The aforementioned strategy allowed us to provide two solutions which overcome the state-of-the-art in the considered aligned scenarios: the first based on k-nearest neighbors ([78]) and the second based on CNN (*CNN-based Estimation of the First Quantization Matrix from Double Com-pressed JPEG Images*, to appear in MDPI Journal of Visual Communication and Image Representation).

Algorithms that can extract a unique fingerprint of digital content have opened up new ways to fight crime. In some cases, the digital content could be an appropriate *digitization* of a real object to authenticate. Moreover, the digitization process could enhance some object's feature, not visible in physical way. An other research track of this PhD was the identification of printed documents: we exploited some descriptors in order to generate a fingerprint from the physical features of the documents. In particular, we exploited the random disposition of the wood fibers that make up the paper sheet: throught the backlight of the physical document this disposition become visible. In this way the method doesn't bind the extraction of the fingerprint to the printed surface of the document (repeatable) but to the physical property of the paper sheet (unique). We studied different descriptors over the three years, building approaches based on them ([101, 102]).

Future works in both the tracks has been scheduled. The method based on CNN for First Quantization Estimation needs to be expanded in the nonaligned scenario; in this case further analyses will be carried out, especially on the understanding of the estimate of the grid shift between first and second compression. The method for real document identification could be analyzed under other descriptors and conditions.

Bibliography

[1] H. Farid. “Image forgery detection”. In: *IEEE Signal processing magazine* 26.2 (2009), pp. 16–25.

[2] P. Blythe and J. Fridrich. “Secure digital camera”. In: *Digital Investigation* (2004).

[3] M. Barni and F. Bartolini. *Watermarking systems engineering: enabling digital assets security and other applications.* Crc Press, 2004.

[4] M. K. Johnson and H. Farid. “Exposing digital forgeries through chromatic aberration”. In: *Proceedings of the 8th workshop on Multimedia and security.* 2006, pp. 48–55.

[5] A. C. Popescu and H. Farid. “Exposing digital forgeries in color filter array interpolated images”. In: *IEEE Transactions on Signal Processing* 53.10 (2005), pp. 3948–3959.

[6] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. “Image forgery localization via fine-grained analysis of CFA artifacts”. In: *IEEE Transactions on Information Forensics and Security* 7.5 (2012), pp. 1566–1577.

[7] J. Fridrich. “Digital image forensics”. In: *IEEE Signal Processing Magazine* 26.2 (2009), pp. 26–37.

[8] J. Lukáš, J. Fridrich, and M. Goljan. “Detecting digital image forgeries using sensor pattern noise”. In: *Security, Steganography, and Watermarking of Multimedia Contents VIII.* Vol. 6072. International Society for Optics and Photonics. 2006, 60720Y.

[9] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva. “Estimate of PRNU noise based on different noise models for source camera identification”. In: *International Journal of Digital Crime and Forensics (IJDCF)* 2.2 (2010), pp. 21–33.

[10] C.-T. Li. “Source camera identification using enhanced sensor pattern noise”. In: *IEEE Transactions on Information Forensics and Security* 5.2 (2010), pp. 280–287.

[11] L. Verdoliva, D. Cozzolino, and G. Poggi. “A feature-based approach for image tampering detection and localization”. In: *2014 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2014, pp. 149–154.

[12] G. K. Wallace. “The JPEG still picture compression standard”. In: *IEEE transactions on consumer electronics* 38.1 (1992), pp. xviii–xxxiv.

[13] Z. Fan and R. L. De Queiroz. “Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history”. In: *Proc. of the International Conference on Image Processing*. 1, 2000, pp. 948–951.

[14] Z. Fan and R. De Queiroz. “Identification of bitmap compression history: JPEG detection and quantizer estimation”. In: *IEEE Trans. on Image Processing* 12.2 (2003), pp. 230–235.

[15] A. R. Bruna, G. Messina, and S. Battiato. “Crop detection through blocking artefacts analysis”. In: *International Conference on Image Analysis and Processing*. Springer. 2011, pp. 650–659.

[16] W. Luo, J. Huang, and G. Qiu. “JPEG error analysis and its applications to digital image forensics”. In: *IEEE Transactions on Information Forensics and Security* 5.3 (2010), pp. 480–491.

[17] D. Fu, Y. Q. Shi, and W. Su. “A generalized Benford’s law for JPEG coefficients and its applications in image forensics”. In: *Security, Steganography, and Watermarking of Multimedia Contents IX*. Vol. 6505. International Society for Optics and Photonics. 2007, p. 65051L.

[18] C. Pasquini, G. Boato, and F. Pérez-González. “Statistical detection of JPEG traces in digital images in uncompressed formats”. In: *IEEE Transactions on Information Forensics and Security* 12.12 (2017), pp. 2890–2905.

[19] F. Benford. “The law of anomalous numbers”. In: *Proceedings of the American philosophical society* (1938), pp. 551–572.

[20] H. Farid. “Exposing digital forgeries from JPEG ghosts”. In: *IEEE transactions on information forensics and security* 4.1 (2009), pp. 154–160.

[21] A. Piva. “An overview on image forensics”. In: *ISRN Signal Processing* 2013 (2013), p. 22. DOI: 10.1155/2013/496701.

[22] L. Verdoliva. “Media forensics and deepfakes: an overview”. In: *IEEE Journal of Selected Topics in Signal Processing* 14.5 (2020), pp. 910–932.

[23] M. C. Stamm, M. Wu, and K. J. R. Liu. “Information Forensics: An Overview of the First Decade”. In: *IEEE Access* 1 (2013), pp. 167–200. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2013.2260814.

[24] B. Li, Y. Q. Shi, and J. Huang. “Detecting doubly compressed JPEG images by using mode based first digit features”. In: *2008 IEEE 10th Workshop on Multimedia Signal Processing*. IEEE. 2008, pp. 730–735.

[25] O. Giudice, F. Guarnera, A. Paratore, and S. Battiato. “1-D DCT Domain Analysis for JPEG Double Compression Detection”. In: *Proc. of International Conference on Image Analysis and Processing*. Springer. 2019, 716–726 (LNCS, volume 11752).

[26] H. Yao, H. Wei, T. Qiao, and C. Qin. “JPEG Quantization Step Estimation with Coefficient Histogram and Spectrum Analyses”. In: *Journal of Visual Communication and Image Representation* (2020), p. 102795.

[27] Z. Lin, J. He, X. Tang, and C.-K. Tang. “Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis”. In: *Pattern Recognition* 42.11 (2009), pp. 2492–2501.

[28] T. Bianchi, A. De Rosa, and A. Piva. “Improved DCT coefficient analysis for forgery localization in JPEG images”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 2444–2447.

[29] Y.-L. Chen and C.-T. Hsu. “Image tampering detection by blocking periodicity analysis in JPEG compressed images”. In: *2008 IEEE 10th Workshop on Multimedia Signal Processing*. IEEE. 2008, pp. 803–808.

[30] T. Bianchi and A. Piva. “Detection of non-aligned double JPEG compression with estimation of primary compression parameters”. In: *2011 18th IEEE International Conference on Image Processing*. IEEE. 2011, pp. 1929–1932.

[31] T. Bianchi and A. Piva. “Detection of nonaligned double JPEG compression based on integer periodicity maps”. In: *IEEE transactions on Information Forensics and Security* 7.2 (2011), pp. 842–848.

[32] Y.-L. Chen and C.-T. Hsu. “Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection”. In: *IEEE Transactions on Information Forensics and Security* 6.2 (2011), pp. 396–406.

[33] T. Bianchi and A. Piva. “Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts”. In: *Proc. of IEEE Trans. on Information Forensics and Security* 7.3 (2012), p. 1003.

[34] M. Barni, M. Fontani, and B. Tondi. “A universal attack against histogram-based image forensics”. In: *International Journal of Digital Crime and Forensics (IJDCF)* 5.3 (2013), pp. 35–52.

[35] M. Barni, M. Fontani, and B. Tondi. “Universal counterforensics of multiple compressed JPEG images”. In: *International Workshop on Digital Watermarking*. Springer. 2014, pp. 31–46.

[36] J. Park, D. Cho, W. Ahn, and H.-K. Lee. “Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 636–652.

[37] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli. “Localization of JPEG double compression through multi-domain convolutional neural networks”. In: *2017 IEEE Conference on computer vision and pattern recognition workshops (CVPRW)*. IEEE. 2017, pp. 1865–1871.

[38] J. Lukáš and J. Fridrich. “Estimation of primary quantization matrix in double compressed JPEG images”. In: *Proc. Digital forensic research workshop*. 2003, pp. 5–8.

[39] J. Dong, W. Wang, T. Tan, and Y. Q. Shi. "Run-length and edge statistics based approach for image splicing detection". In: *International workshop on digital watermarking.* Springer. 2008, pp. 76–87.

[40] K. Bahrami, A. C. Kot, L. Li, and H. Li. "Blurred image splicing localization by exposing blur type inconsistency". In: *IEEE Transactions on Information Forensics and Security* 10.5 (2015), pp. 999–1009.

[41] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. "A sift-based forensic method for copy–move attack detection and transformation recovery". In: *IEEE transactions on information forensics and security* 6.3 (2011), pp. 1099–1110.

[42] D. Cozzolino, G. Poggi, and L. Verdoliva. "Efficient dense-field copy–move forgery detection". In: *IEEE Transactions on Information Forensics and Security* 10.11 (2015), pp. 2284–2297.

[43] M. Barni, Q.-T. Phan, and B. Tondi. "Copy move source-target disambiguation through multi-branch CNNs". In: *IEEE Transactions on Information Forensics and Security* 16 (2020), pp. 1825–1840.

[44] Y. Rao and J. Ni. "A deep learning approach to detection of splicing and copy-move forgeries in images". In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS).* IEEE. 2016, pp. 1–6.

[45] L. Guarnera, O. Giudice, C. Nastasi, and S. Battiato. "Preliminary forensics analysis of deepfake images". In: *2020 AEIT International Annual Conference (AEIT).* IEEE. 2020, pp. 1–6.

[46] L. Guarnera, O. Giudice, and S. Battiato. "Deepfake detection by analyzing convolutional traces". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.* 2020, pp. 666–667.

[47] L. Guarnera, O. Giudice, and S. Battiato. "Fighting deepfake by exposing the convolutional traces on images". In: *IEEE Access* 8 (2020), pp. 165085–165098.

[48] O. Giudice, L. Guarnera, and S. Battiato. "Fighting deepfakes by detecting GAN DCT anomalies". In: *arXiv preprint arXiv:2101.09781* (2021).

[49] A. Jain, P. Majumdar, R. Singh, and M. Vatsa. "Detecting GANs and retouching based digital alterations via DAD-HCNN". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 672–673.

[50] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato. "A classification engine for image ballistics of social data". In: *International Conference on Image Analysis and Processing*. Springer. 2017, pp. 625–636.

[51] G. K. Wallace. "The JPEG still picture compression standard". In: *Communications of the ACM* 34.4 (1991), pp. 30–44.

[52] B. Li, T. Ng, X. Li, S. Tan, and J. Huang. "Statistical model of JPEG noises and its application in quantization step estimation". In: *IEEE Trans. on Image Processing* 24.5 (2015), pp. 1471–1484.

[53] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato. "First quantization matrix estimation from double compressed JPEG images". In: *IEEE Trans. on Information Forensics and Security* 9.8 (2014), pp. 1299–1310.

[54] N. Dalmia and M. Okade. "First quantization matrix estimation for double compressed JPEG images utilizing novel DCT histogram selection strategy". In: *Proc. of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. 2016, pp. 1–8.

[55] N. Dalmia and M. Okade. "Robust first quantization matrix estimation based on filtering of recompression artifacts for non-aligned double compressed JPEG images". In: *Signal Processing: Image Communication* 61 (2018), pp. 9–20.

[56] F. Xue, Z. Ye, W. Lu, H. Liu, and B. Li. "MSE period based estimation of first quantization step in double compressed JPEG images". In: *Signal Processing: Image Communication* 57 (2017), pp. 76–83.

[57] T. H. Thai, R. Cogranne, and F. Retraint. "Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm". In: *IEEE Transactions on Image Processing* 23.5 (2014), pp. 1980–1993.

[58] T. H. Thai and R. Cogranne. “Estimation of primary quantization steps in double-compressed JPEG images using a statistical model of discrete cosine transform”. In: *IEEE Access* 7 (2019), pp. 76203–76216.

[59] G. Varghese and A. Kumar. “Detection of Double JPEG Compression on Color Image using Neural Network Classifier”. In: *International Journal for Innovative Research in Science and Technology* 3.3 (2016), pp. 175–181.

[60] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. “Aligned and non-aligned double JPEG detection using convolutional neural networks”. In: *Journal of Visual Communication and Image Representation* 49.Supplement C (2017), pp. 153 –163. ISSN: 1047-3203.

[61] T. Uricchio, L. Ballan, R. Caldelli, and I. Amerini. “Localization of JPEG Double Compression Through Multi-Domain Convolutional Neural Networks”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 53–59.

[62] Q. Wang and R. Zhang. “Double JPEG compression forensics based on a convolutional neural network”. In: *EURASIP Journal on Information Security* 2016.1 (2016), p. 23.

[63] Y. Niu, B. Tondi, Y. Zhao, and M. Barni. “Primary Quantization Matrix Estimation of Double Compressed JPEG Images via CNN”. In: *IEEE Signal Processing Letters* 27 (2020), pp. 191–195.

[64] B. Tondi, A. Costanzo, D. Huang, and B. Li. “Boosting CNN-based primary quantization matrix estimation of double JPEG images via a classification-like architecture”. In: *EURASIP Journal on Information Security* 2021.1 (2021), pp. 1–14.

[65] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi. “Computational Data Analysis for First Quantization Estimation on JPEG Double Compressed Images”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 5951–5958.

[66] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi. "Estimating Previous Quantization Factors on Multiple JPEG Compressed Images". In: *EURASIP Journal on Information Security* 2021.8 (2021), pp. 1–11. DOI: 10.1186/s13635-021-00120-7.

[67] G. Hudson, A. Léger, B. Niss, and I. Sebestyén. "JPEG at 25: Still going strong". In: *IEEE MultiMedia* 24.2 (2017), pp. 96–103.

[68] D. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. "RAISE: a raw images dataset for digital image forensics". In: *Proc. of the 6th ACM Multimedia Systems Conference*. 2015, pp. 219–224.

[69] P. Bas, T. Filler, and T. Pevnỳ. "Break our steganographic system: the ins and outs of organizing BOSS". In: *International workshop on information hiding*. Springer. 2011, pp. 59–70.

[70] T. H. Thai, R. Cogranne, F. Retraint, and T. Doan. "JPEG quantization step estimation and its applications to digital image forensics". In: *IEEE Trans. on Information Forensics and Security* 12.1 (2016), pp. 123–133.

[71] T. Gloe and R. Böhme. "The Dresden image database for benchmarking digital image forensics". In: *Journal of Digital Forensic Practice* 3.2-4 (2010), pp. 150–159.

[72] G. Schaefer and M. Stich. "UCID: An uncompressed color image database". In: *Storage and Retrieval Methods and Applications for Multimedia 2004*. Vol. 5307. International Society for Optics and Photonics. 2003, pp. 472–480.

[73] J. Park, D. Cho, W. Ahn, and H. Lee. "Double JPEG Detection in Mixed JPEG Quality Factors using Deep Convolutional Neural Network". In: *The European Conference on Computer Vision (ECCV)*. 2018.

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

[75] E. Kee, M. K. Johnson, and H. Farid. "Digital image authentication from JPEG headers". In: *IEEE Trans. on Information Forensics and Security* 6.3 (2011), pp. 1066–1075.

[76] S. Agarwal and H. Farid. "Photo forensics from JPEG dimples". In: *2017 IEEE Workshop on Information Forensics and Security (WIFS).* IEEE. 2017, pp. 1–6.

[77] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi. "In-Depth DCT Coefficient Distribution Analysis for First Quantization Estimation". In: *MM-ForWild2020 – MultiMedia FORensics in the WILD 2020 - Workshop of 25th International Conference on Pattern Recognition, ICPR.* Springer. 2020, (LNCS, volume 12666).

[78] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi. "First Quantization Estimation by a Robust Data Exploitation Strategy of DCT Coefficients". In: *IEEE Access* 9 (2021), pp. 73110–73120. DOI: 10.1109/ACCESS.2021.3080576.

[79] E. Y. Lam and J. W. Goodman. "A mathematical analysis of the DCT coefficient distributions for images". In: *IEEE Trans. on Image Processing* 9.10 (2000), pp. 1661–1666.

[80] 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.

[81] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt. "Combating digital document forgery using new secure information hiding algorithm". In: *International Conference on Digital Information Management.* 2008, pp. 922–924. DOI: 10.1109/ICDIM.2008.4746807.

[82] A. G. H. Ahmed and F. Shafait. "Forgery Detection Based on Intrinsic Document Contents". In: *International Workshop on Document Analysis Systems.* 2014, pp. 252–256. DOI: 10.1109/DAS.2014.26.

[83] A. C. Berenguel, O. R. Terrades, J. C. Lladós, and C. M. Cañero. "Banknote Counterfeit Detection through Background Texture Printing Analysis". In: *IAPR Workshop on Document Analysis Systems.* 2016, pp. 66–71. DOI: 10.1109/DAS.2016.34.

[84] A. R. Bruna, G. M. Farinella, G. C. Guarnera, and S. Battiato. "Forgery Detection and Value Identification of Euro Banknotes". In: *Sensors* 13.2 (2013), pp. 2515–2529. DOI: 10.3390/s130202515.

[85] N. K. Gill, R. Garg, and E. A. Doegar. “A review paper on digital image forgery detection techniques”. In: *International Conference on Computing, Communication and Networking Technologies*. 2017, pp. 1–7. DOI: 10.1109/ICCCNT.2017.8203904.

[86] M. Kumar, S. Gupta, and N. Mohan. “A computational approach for printed document forensics using SURF and ORB features”. In: *Soft Computing* (2020). DOI: 10.1007/s00500-020-04733-x.

[87] A. C. Berenguel, O. R. Terrades, J. C. Lladós, and C. M. Cañero. “Identity Document and banknote security forensics: a survey”. In: *arXiv preprint 1910.08993* (2019). arXiv: 1910.08993 [cs.CV].

[88] 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*. 2016, pp. 5–16.

[89] W. Samsul, H. P. Uranus, and M. D. Birowosuto. “Recognizing Document’s Originality by laser Surface Authentication”. In: *International Conference on Advances in Computing, Control and Telecommunication Technologies*. 2010, pp. 37–40. DOI: 10.1109/ACT.2010.15.

[90] A. Sharma, L. Subramanian, and E. A. Brewer. “PaperSpeckle: microscopic fingerprinting of paper”. In: *ACM Conference on Computer and Communications Security*. 2011, pp. 99–110. DOI: 10.1145/2046707.2046721.

[91] J. D. Buchanan, R. P. Cowburn, A.-V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, and M. T. Bryan. “Forgery: Fingerprinting documents and packaging”. In: *Nature* 43.28 (2005).

[92] F. Van Beijnum, E. G. Van Putten, K. L. Van der Molen, and A. P. Mosk. “Recognition of paper samples by correlation of their speckle patterns”. In: *arXiv preprint physics/0610089* (2006).

[93] R. Cowburn. “Laser Surface Authentication - natural randomness as a fingerprint for document and product authentication”. In: *Optical Document Security Conference*. 2008.

[94] R. Cowburn. “Laser surface authentication – reading Nature’s own security code”. In: *Contemporary Physics* 49.5 (2008), pp. 331–342. DOI: 10.1080/00107510802583948.

[95] W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J. A. Halderman, and E. W. Felten. “Fingerprinting blank paper using commodity scanners”. In: *IEEE Symposium on Security and Privacy*. 2009, pp. 301–314. DOI: 10.1109/SP.2009.7.

[96] E. Toreini, S. F. Shahandashti, and F. Hao. “Texture to the Rescue: Practical Paper Fingerprinting based on Texture Patterns”. In: *ACM Transactions on Privacy and Security* 20.3 (2017), pp. 1–29. DOI: 10.1145/3092816.

[97] C.-W. Wong and M. Wu. “Counterfeit Detection Based on Unclonable Feature of Paper Using Mobile Camera”. In: *IEEE Transactions on Information Forensics and Security* 12.8 (2017), pp. 1885–1899. DOI: 10.1109/TIFS.2017.2694404.

[98] R. Liu, C.-W. Wong, and M. Wu. “Enhanced Geometric Reflection Models for Paper Surface Based Authentication”. In: *IEEE International Workshop on Information Forensics and Security*. 2019. DOI: 10.1109/WIFS.2018.8630759.

[99] D. Chen, Q. Hu, and S. Zeng. “An Anti-Counterfeiting Method of High Security and Reliability Based on Unique Internal Fiber Pattern of Paper”. In: *2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*. IEEE, pp. 174–178.

[100] T. Haist and H. J. Tiziani. “Optical detection of random features for high security applications”. In: *Optics Communications* 147.1 (1998), pp. 173–179. DOI: 10.1016/S0030-4018(97)00546-4.

[101] F. Guarnera, D. Allegra, O. Giudice, F. Stanco, and S. Battiato. “A New Study On Wood Fibers Textures: Documents Authentication Through LBP Fingerprint”. In: *IEEE International Conference on Image Processing*. 2019, pp. 4594–4598. DOI: 10.1109/ICIP.2019.8803502.

[102] F. Guarnera, O. Giudice, D. Allegra, F. Stanco, S. Battiato, S. Livatino, V. Matranga, and A. Salici. “A Robust Document Identification Framework through f-BP Fingerprint”. In: *Journal of Imaging* 7.8 (2021), p. 126.

[103] S. Brahnam, L. C. Jain, L. Nanni, and A. Lumini, eds. *Local Binary Patterns: New Variants and Applications*. Springer-Verlag Berlin Heidelberg, 2014. ISBN: 978-3-642-39289-4. DOI: 10.1007/978-3-642-39289-4.

[104] T. Ojala, M. Pietikäinen, and D. Harwood. "A comparative study of texture measures with classification based on featured distributions". In: *Pattern Recognition* 29.1 (1996), pp. 51–59. DOI: 10.1016/0031-3203(95)00067-4.

[105] A. Bhattacharyya. "On a measure of divergence between two statistical populations defined by their probability distributions". In: *Bulletin of the Calcutta Mathematical Society* 35 (1943), pp. 99–109.

[106] X. Tan and B. Triggs. "Enhanced local texture feature sets for face recognition under difficult lighting conditions". In: *Analysis and Modeling of Faces and Gestures*. 2007, pp. 168–182. DOI: 10.1007/978-3-540-75690-3_13.

[107] T. P. Nguyen, N.-S. Vu, and A. Manzanera. "Statistical binary patterns for rotational invariant texture classification". In: *Neurocomputing* 173 (2016), pp. 1565–1577. DOI: 10.1016/j.neucom.2015.09.02.

[108] Z. Guo, L. Zhang, and D. Zhang. "A completed modeling of local binary pattern operator for texture classification". In: *IEEE Transactions on Image Processing* 19.6 (2010), pp. 1657–1663. DOI: 10.1109/TIP.2010.2044957.

[109] K. Pearson. "On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine* 2.6 (1901), pp. 559–572.