UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Statistical and deterministic approaches for multimedia forensics

Cecilia Pasquini

Advisor:

Prof. Giulia Boato,

Università degli Studi di Trento

March 2016

Abstract

The increasing availability and pervasiveness of multimedia data in our society is before our very eyes. As a result of globalization and worldwide connectivity, people from all over the planet are exchanging constantly increasing amounts of images, videos, audio recordings on a daily basis. Coupled with the easy access to user-friendly editing software, this poses a number of problems related to the reliability and trustworthiness of such content, as well as its potential malevolent use. For this reason, the research field of multimedia forensics focuses on the development of forensic tools for verifying the authenticity of multimedia data. The hypothesis of pristine status of images, videos or audio tracks is called into question and can be rejected if traces of manipulation are detected with a certain degree of confidence. In this framework, studying traces left by any operation that could have been employed to process the data, either for malicious purposes or simply to improve their content or presentation, turns out to be of interest for a comprehensive forensic analysis.

The goal of this doctoral study is to contribute to the field of multimedia forensics by exploiting intrinsic statistical and deterministic properties of multimedia data.

With this respect, much work has been devoted to the study of JPEG compression traces in digital images, resulting in the development of several innovative approaches. Indeed, some of the main related research problems have been addressed and solution based on statistical properties of digital images have been proposed. In particular, the problem of identifying traces of JPEG compressions in images that have been decompressed and saved in uncompressed formats has been extensively studied, resulting in the design of novel statistical detectors. Given the enormous practical relevance, digital images in JPEG formats have also been considered. A novel method aimed at discriminating images compressed only once and more than once has been developed, and tested on a variety of images and forensic scenarios. Being the potential presence of intelligent counterfeiters ever increasingly studied, innovative counterforensic techniques to JPEG compression based on smart reconstruction strategies are proposed.

Finally, we explore the possibility of defining and exploiting deterministic properties related to a certain processing operation in the forensic analysis. With this respect, we present a first approach targeted to the detection in one-dimensional data of a common data smoothing operation, the median filter. A peculiarity of this method is the ability of providing a deterministic response on the presence of median filtering traces in the data under investigation.

Keywords
Multimedia forensics, JPEG compression forensics, counterforensics, deterministic analysis, median filter.

Contents

Chapter 1

Introduction

"What is essential is invisible to the eye"
A. de Saint-Exupéry

Multimedia objects have become more and more pervasive in our society. This is mainly due to the increasing number of devices able to take pictures and record audio-video tracks, such as low-cost digital cameras, smartphones or tablets. At the same time, modifications and manipulations of such multimedia content can nowadays be performed by non-expert users, thanks to the availability of user-friendly software tools. Finally, the easy access to sharing platforms offered by the web and the cloud technologies (such as social networks, blogs, online newspapers) allows users to instantaneously broadcast multimedia objects that might be significantly modified.
As a result, the trustworthiness of media contents is strongly compromised, with effects that span in many practical scenarios. Considering also the more intuitive and immediate impact of visual data with respect to textual documents, the potential diffusion of distorted or completely fake multimedia content on websites, information media, advertisement and legal proceedings seriously represents an issue to be addressed. It appears evident that the authenticity of multimedia data can no longer be taken from granted, and their fidelity to reality should be called into question. In other words, the development of media-related technologies has to be combined with effective techniques for their protection and verification. This would help in avoiding an illegitimate exploitation, be it malicious or not, of the semantic message they may convey. Indeed, the role of visual content manipulation is currently under debate, questioning how manipulated images impact users' perceptions and opinions on topics and people [37, 143].

We report in Fig. 1.1 some edited images, serving as examples of the implications that superimposed visual changes can have when shared on a large scale. Fig. 1.1a contains the original image of the construction site of the hydroelectric dam of Belo Monte in Brazil, while Fig. 1.1b contains its digitally modified version, published on *The Spiegel* newspaper in 2013 in an article evaluating the environmental impact of the structure[1]. The visual impact of the image is clearly altered, as the site looks more degraded due to

[1] www.spiegel.de/international/world/growing-concern-that- news-photos-are-being-excessively-manipulated-a-898509.html

Figure 1.1: Example of modified images on the web.

the social and environmental consequences of the construction of the dam. An example involving international politics concerns the march featuring state leaders from all over the world that was organized in Paris after the Charlie Hebdo terroristic attack in January 2015. An orthodox paper published a modified image (Fig. 1.1d) where female leaders were deliberately edited out from the original image (Fig. 1.1c)[2]. A further example is reported in Fig. 1.1e and 1.1f: the modified picture on the right was published on Twitter by the 10 Downing Street profile, revealing the superimposition of a Remembrance Sunday poppy (traditionally exposed at the day of publication to commemorate fallen

[2]http://www.mirror.co.uk/news/world-news/charlie-hebdo-female-world-leaders-4976457

soldiers during World War I) onto David Cameron's lapel. After this was spotted in the first place, the episode went viral on the web and social networks[3].

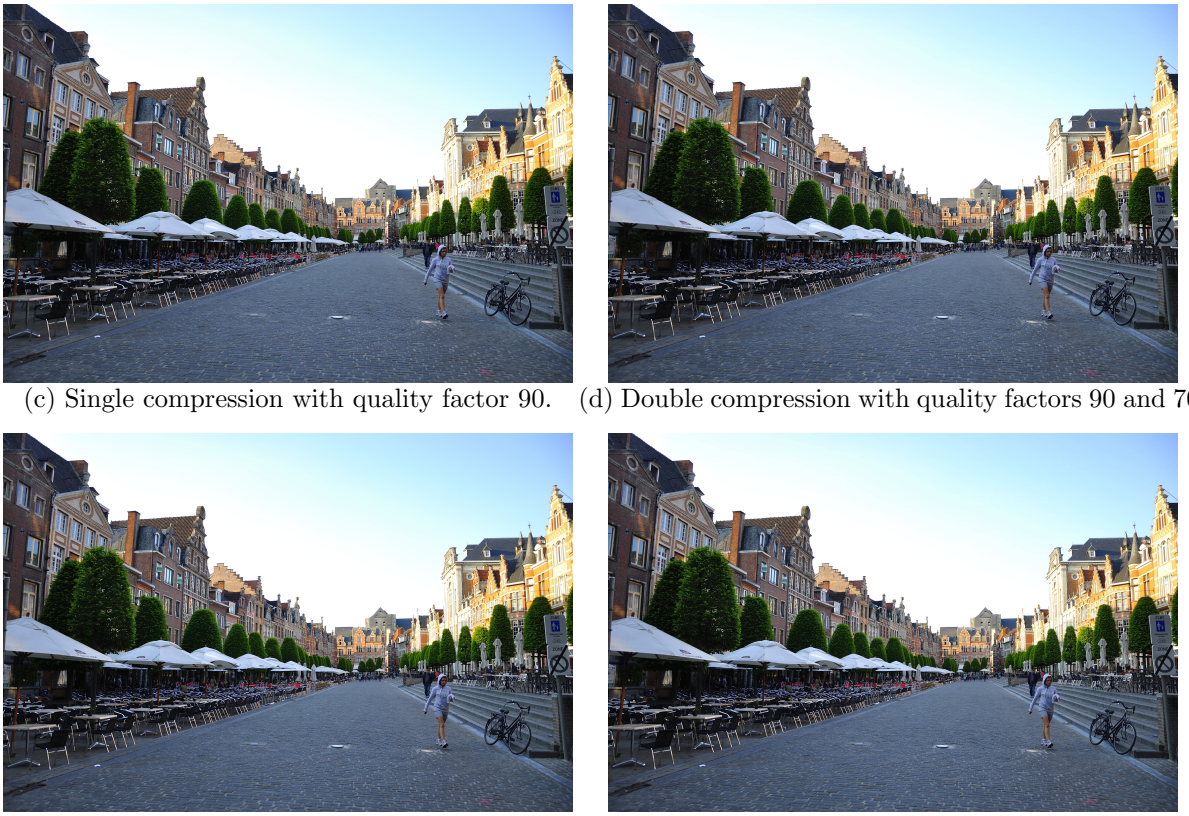
As a matter of fact, the verification and authentication of information, including multimedia content, in web and social media is drawing increasing attention from a journalistic and enterprise perspective [1]. Moreover, nowadays multimedia objects are strongly present in modern digital investigations, where digital images, audio tracks and video sequences more and more frequently represent potential digital evidences to the court [119]. In this scenario, in addition to the analysis of physical machines (i.e., computer forensics [23]), it is necessary to perform a forensic analysis of the multimedia data itself, in order to assess the reliability of the content.
Driven by such motivations, in the last decade the scientific community has been developing a constantly increasing amount of techniques targeted to the authentication of multimedia objects, under the name of *digital multimedia forensics* [125]. Differently from active techniques like digital watermarking, where an imperceptible digital code (a watermark) is inserted into a multimedia object before its delivering/sharing, in multimedia forensics we do not assume such a priori information. Indeed, this would imply the use of special equipped devices when creating the multimedia content, whereas in a general scenario data generated by a large number of different devices need to be analyzed. For this reason, forensic methods are said to be *passive* and they are generally based on the following principle: *manipulations of the multimedia signal may be visually unperceivable but leave traces that can be detected by means of proper forensic methodologies.*

In particular, great attention has been paid to digital image forensics for which a wide variety of different processing have been studied [51]. In this framework, the traces left by JPEG compression, the most popular coding scheme for images, have been widely analyzed and exploited in different forensic scenarios. Indeed, there are methods aimed at detecting traces of previous compression in uncompressed images, double compression in compressed images, inconsistencies within the same image. As other kind of processing, a single or repeated JPEG compression performed with reasonably high quality level generally does not leave evident traces from a perceptual perspective (especially in high resolution images), thus excluding the possibility of a preliminary visual inspection. In Fig. 1.2, we report the same image processed according to different compression chains, as a demonstration of the generally unperceivable visual effect of one or more compressions. Thus, in absence of *a priori* information or metadata, the only way to determine whether previous compressions occurred is to analyze the *underlying statistics* of the two-dimensional signal, from which potential anomalies can be revealed. With this respect, a number of notable approaches have been proposed in the literature, although most of the them still present significant limitations in terms of detection capabilities and robustness to diverse kind of images.

In addition, as it happened to digital watermarking and steganography, the need of an adversary-aware perspective recently emerged also in multimedia forensics. Indeed, while forensic methods are quite effective in case of unrefined manipulations, the poten-

[3] http://www.theguardian.com/politics/2015/nov/02/poppy-photoshopped-david-cameron-facebook-picture

(a) No compression (b) Single compression with quality factor 70.



(c) Single compression with quality factor 90. (d) Double compression with quality factors 90 and 70.

Figure 1.2: Different chains of compression are applied to the same image.

tial presence of a smart adversary compromises the reliability of current techniques: recent research has shown that they can be deceived if the forger is aware of the forensic tools and adopts specific countermeasures, namely *counter-forensic* (or *anti-forensic*) attacks [76].

Several counter-forensic methods have been proposed, with particular attention to the case of JPEG compression. Research in this direction helped in highlighting drawbacks and limitations of forensic methods and, consequently, in assessing their reliability. As a matter of fact, the expression *adversarial signal processing* [12] has been introduced, referring to any branch of signal processing that is conceived in an adversary-aware perspective, such as watermarking, spam filtering, secure machine learning, network intrusion detection, anti-spoofing biometrics.

1.1 Novel solutions and innovative aspects

In this doctoral study, we propose contributions to the forensic task of detecting certain processing operations in multimedia data, with particular attention to digital images. The general rationale behind our work is to leverage statistical and deterministic properties that are common to the kind of data under investigation. As it will be explored in detail, this helps the forensic analysis in several directions, like assessing the confidence of a certain decision or avoiding preliminary training phases by exploiting closed-form theoretical derivations.
The main contributions of our work is detailed as follows:

- **Detection of JPEG compression traces in uncompressed format images.** Intrinsic statistical properties of natural uncompressed images are used in image forensics for detecting traces of previous processing operations. We perform a theoretical analysis of Benford–Fourier coefficients (BF) computed on the 8×8 block-DCT domain, originally proposed in [109]. The distribution of such coefficients is derived theoretically both under the hypotheses of no compression and previous compression with a certain quality factor, allowing also for the computation of the respective likelihood functions. Then, three classification tests based on different statistics are proposed, relying on discriminative thresholds that can be determined automatically, without the need of any training phase. The statistical analysis is based on the assumption of Generalized Gaussian distribution of DCT coefficients, which generally holds for any uncompressed image. As a result, the method proves to be suitable for images of different size and source camera. Experiments on real images and comparisons with state-of-art techniques show that the proposed approach outperforms existing ones and overcomes issues due to dataset-dependency.

- **Detection of multiple JPEG compression traces in JPEG images.** The analysis of BF coefficients is extended to the case of JPEG images, resulting in a forensic method for the identification of multiple aligned JPEG compressions and the estimation of the corresponding quality factors. It is based on the computation of likelihood function values for the null hypothesis of single compression and the alternative hypothesis of multiple compression with a certain sequence of quantization tables. In principle, the technique allows for the detection of an arbitrary number of compressions, depending on the pool of the tested alternative hypotheses. Experimental results show that the performance of the proposed technique is good also in the case of last compression heavier than the previous ones, where existing methods usually lead to high false negative rates. Moreover, the entire chain of JPEG compression is reconstructed also in case of triple compressed images, while existing methods generally estimate only one previous quality factor applied.
  The effectiveness of the method is also explored in the challenging case of double high quality JPEG compression, for which specific improvements are adopted. In this case, the detection scheme needs to be modified to avoid high false alarm rates, and complementary techniques are integrated to perform a comprehensive analysis.

- **Counterforensics of JPEG images.** Two counterforensic techniques are proposed for the reconstruction of statistical properties of natural and JPEG images. They both target the modification of the First Significant Digit (FSD) histogram of the DCT coefficients, in order to conceal traces of single and, in some cases, multiple compression. The first one operates in the FSD modular logarithmic domain, from which FSD and DCT coefficients distribution typical of uncompressed images are consequently reconstructed. It is applied to single compressed JPEG images and restores the Gaussian-like statistical distribution of DCT coefficients and the Benford's law distribution of the FSD randomization strategy in a specific domain, the method is compared with a well-established existing anti-forensic attack in terms of quality of the resulting image. In addition, the effectiveness of our approach as counter-forensics processing is assessed by measuring its impact on the performance of two different forensic tools.
  On the other hand, the second one directly targets the reconstruction of a given FSD histogram and can be seen as universal to detectors based on FSD first-order histogram. Based on heuristic criteria, the technique provides a close-to-optimal solution for the problem of FSD histogram modification with minimal distortion in terms of Mean Square Error (MSE) distortion. The problem is expressed as a two-step optimization process and the proposed solution is tested in a more general forensic scenario where statistics after an arbitrary number of compressions is targeted, including comparison with state-of-the-art similar techniques.

- **Deterministic detection of median filtering in data sequences.** This work represents our first attempt to define and leverage properties that are deterministically related to a certain processing, in contrast with typical forensic methodologies based on statistical properties. We propose a forensic technique able to detect the application of a median filter to 1D data. Relying on deterministic mathematical properties of the median filter, we identify specific order relationships among the sample values that cannot be found in filtered sequences. Hence, their presence in the analyzed 1D sequence allows excluding the application of the median filter. Due to its deterministic nature, the method ensures a null false negative rate and, although false positives (not filtered sequences classified as filtered) are theoretically possible, experimental results show that the false alarm rate is null for sufficiently long sequences. Furthermore, the proposed technique has the capability to locate with good precision a median filtered part of 1D data and provides a good estimate of the window size used.

1.2 Outline

The remaining part of the thesis is structured as follows.
In Chapters 2, we introduce the field of multimedia forensics and an overview of the existing techniques based on JPEG compression traces.
The first contribution on the detection of JPEG compression traces in uncompressed format images is presented in Chapter 3.
The analysis of JPEG format images and the detection scheme of multiple compression traces are reported in Chapter 4, while we deal with the case of high quality repeated compression in Chapter 5.
In Chapter 6, novel counterforensic techniques for JPEG compression are presented.
Chapter 7 proposes the methodology designed to detect median filter in one-dimensional data sequences based on deterministic properties of the filter.
Finally, conclusions are drawn in Chapter 8.

Publications

The research presented in this dissertation resulted in the following publications:

Chapter 3:

C. Pasquini, F. Pèrez-Gonzàlez, and G. Boato. A Benford-Fourier JPEG compression detector. In *IEEE International Conference on Image Processing (ICIP)*, pages 5322-5326, 2014. [106]

C. Pasquini, G. Boato and F. Pèrez-Gonzàlez. Statistical detection of JPEG traces in digital images in uncompressed formats. Submitted to *IEEE Transactions on Information Forensics and Security*. [103]

Chapter 4:

C. Pasquini, G. Boato, and F. Pèrez-Gonzàlez. Multiple JPEG compression detection by means of Benford-Fourier coefficients. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 113-118, 2014. [102]

Chapter 5:

C. Pasquini, P. Schöttle, R. Böhme, G. Boato, F. Pèrez-Gonzàlez. Forensics of high quality and nearly identical JPEG image recompression. Accepted to appear in *ACM Conference in Information Hiding & Multimedia Security (IH&MMSec)*, 2016. [107]

Chapter 6:

C. Pasquini and G. Boato. JPEG compression anti-forensics based on first significant digit distribution. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 500-505, 2013. *Top 10% paper award*. [100]

C. Pasquini, P. Comesaña-Alfaro, F. Pèrez-Gonzàlez, G. Boato. Transportation-theoretic image counterforensics to First Significant Digit histogram forensics. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2718-2722, 2014. [105]

Chapter 7:

C. Pasquini, G. Boato, N. Anjalic, F.G.B. De Natale. A deterministic approach to detect median filtering in 1D data, *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 7, pp. 1425-1437, 2016. [101]

The following papers were published during the course of the PhD but not included in this thesis:

C. Pasquini, C. Brunetta, A. Vinci, V. Conotter and G. Boato. Towards the verification of image integrity in online news. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1-6, 2015. [104]

D.T. Dang-Nguyen, **C. Pasquini**, V. Conotter and G. Boato. RAISE - A raw images dataset for digital image forensics. In *ACM Multimedia Systems Conference (MMSys)*, pp. 219-224, 2015. [44]

Chapter 2

Multimedia forensics and JPEG compression forensics

In this chapter, we present a brief overview of the field of multimedia forensics, illustrating the different kind of problems arising and methodologies proposed, as well as highlighting the main open research directions. Then, we report a review of forensic and counterforensic techniques devoted to JPEG images and the analysis of their digital history. In particular, existing methods are divided according to the forensic task they target and their main limitations are discussed, thus representing the starting point for the novel approaches developed in this doctoral study. Finally, the benchmarking datasets used throughout the thesis are presented.

2.1 Digital multimedia forensics: advances and limitations

Multimedia forensics focuses on verifying the authenticity of audio-visual content, by determining its processing history and origin. Differently from approaches as digital watermarking, no a priori knowledge on the object under investigation is assumed, meaning that the forensic analyst does not look for a specific signature. On the other hand, she should consider a variety of potential manipulations occurred.

We can trace back the origins of multimedia forensics to the seminal work reported in [49], where the idea of detecting manipulations in digital multimedia objects by means of mathematical models was introduced. Since then, a variety of forensic scenarios have been targeted and methods facing specific tasks have been developed. Moreover, a fascinating aspect of multimedia forensics is its strongly interdisciplinary nature. Indeed, application requirements has brought researchers to connect and jointly exploit methodologies typically used in different research fields, like signal processing, computer science, machine learning, game theory. This results in a wide variety of existing tools, for which rigorous classifications and boundaries are hard to determine.

A number of survey papers and books are available and propose different ways of grouping the methods available [51, 122, 121, 117, 65, 125, 111]. We choose to consider as a first distinction the following four main research areas [121] [125]: *tampering detection*, *counterforensics*, *source identification* and *discrimination between CG and natural content*. In

the following, we give an overview of the main goals and approaches for each of them. The list of contributions mentioned is by no means exhaustive, but we address the reader to the overview papers mentioned above and the references therein for a comprehensive review. Moreover, since the research activity within this doctoral study mainly focused on the tampering detection and counterforensics areas, a special attention is devoted to them.

- **Tampering detection**: the goal is to determine the presence or the absence of forgery traces in the content under investigation, thus gathering information on its authenticity. Given the diverse kinds of processing a multimedia object can undergo, many methods have been proposed to detect different types of forgeries, by exploiting various features of the analyzed object.
  In order to give an overview of the existing tools, we refer to the classification presented in [51] for images, keeping in mind that it can be extended to techniques for audio-visual content in general [90]. A very wide class is represented by the *pixel-based* methods, designed for specific forensic problems. This group of techniques study the statistical behaviour of pixel values and identify anomalies or unexpected periodicities produced when the image is altered. They can be applied to detect different forgeries and are particularly efficient in case of copy-move [83, 8, 40, 5] (portion of an image is copy and pasted in the same scene) and splicing [96, 41, 41, 142] (portion of an image is inserted in another one), or resampling [133, 130, 52], a very common process occurring anytime an image is resized. *Camera-based* techniques rely on the assumption that any recording device leaves an intrinsic signature in a image and any forgery generates inconsistencies in chromatic aberration phenomena, color filter array patterns or sensor noise (PRNU) statistical patterns. Such methods are often used for source identification problem [87], as described below.
  Unfortunately, the traces used by the classes described above are strongly compromised when a lossy coding scheme is applied to the image, like in the case of JPEG, and only few methods are robust enough to be reliable also after a compression. Being the most commonly used image format, the analysis of JPEG images plays a key role in the forensic analysis and many efforts have been addressed to the study of statistical traces left by such coding scheme. This leads to another class of methods, the *format-based* ones, where unique properties of coding schemes are used in different forensic decision problems. Finally, *physics-based* and *geometric-based* techniques are proposed. In these approaches, the image is decomposed by means of 2D or 3D models and potential forgeries are exposed by detecting inconsistencies in the distribution of light [28] or in the object projection geometry with respect to the camera [38, 73, 68].

- **Counterforensics.** Literature on digital image forensics rapidly developed in the last decade and efficient methods for diverse kinds of manipulation are available. The situation dramatically changes when a smart counterfeiter is aware of the forensic tools and adopts specific countermeasures. The possibility of skilled adversaries able to deceive the forensic tools was pointed out in [63] and formalized in [14, 123], drawing a growing attention among researchers and currently representing a very

active topic.
Counter-forensic techniques have been proposed for defeating specific pixel-based, camera-based and format-based forensic methods by concealing the traces of previous manipulations, and prove to have a very strong impact on the performance of the forensic detectors they target. For instance, methodologies have been proposed for defeating resampling detectors [74], median filtering detectors [55], camera-based methods [63], lossy compression detectors [124], CFA-based techniques [75], histogram-based methods [11, 43].

- **Source identification.** Such branch of multimedia forensics aims at establishing a link between an object and the device it was acquired with (e.g., camera, mobile phone, scanner). This is done by characterizing noise-like patterns overlaid onto the object during the acquisition process. A number of effective methodologies has been proposed in the literature, exploiting chromatic aberration phenomena [69], demosaicing artifacts [26], sensor defections [45], photo-response non-uniformity noise (PRNU) [29, 82]. An overview of related approaches can be found in [87].

- **Discrimination between CG and natural content**

  Modern computer graphics technologies are nowadays able to produce extremely realistic content, forcing researchers to develop methods to passively distinguish natural scenes from computer generated (CG) ones. Techniques targeting the identification of CG characters and faces have been proposed, exploiting different features of the content under investigation. In addition to geometrical and statistical properties of natural scenes [46, 97], motion and lighting patterns have been used to discriminate faces of CG characters [42], as well as physiological signals [35].

2.1.1 Open problems and research directions

Although a wide literature is available for different multimedia forensic scenarios, the available methodologies still present relevant open issues. In the following, we enumerate some of the general limitations of the current state-of-the-art in multimedia forensics tampering detection. Besides currently compromising the actual applicability of existing tools in real situations, such flaws represent stimulating challenges for the scientific community, which is intensively working towards these directions.

- **Lack of theoretical models**: a relevant issue is the fact that many forensic tools are based on statistical properties for which theoretical models are not available. Although they usually achieve excellent results in certain experimental settings, the absence of a generalized model might result in non-controllable performance when the test setting is modified, since the parameters of the methods change as well. This generally happens, for instance, for approaches implying the need of machine learning techniques. Although they represent extremely useful tools and are able to automatically capture statistical patterns, they usually require training phases that are not always feasible. Moreover, they suffer from typical automatic learning issues (as overfitting or dataset-dependency), which might have a strong impact on the applicability of multimedia forensic techniques in different fields. On the other hand,

a closed-form statistical model of the quantities involved would help in assessing the confidence of the results obtained. It would also allow to design hypothesis tests with confidence intervals determined theoretically, instead of relying on dataset-dependent thresholds.
As an example, we consider the class of format-based methods for JPEG images based on the distribution of First Significant Digits (FSD) of DCT coefficients. In [59], a generalized version of Benford's law is proposed and in [81] the distance with respect to such distribution is exploited for discriminating single and double compressed images. In [91], the histogram of FSDs is also used for identifying the number of previous compressions by means of a combination of binary SVM classifiers. However, at the current state there is no theoretical explanation for the behavior of FSDs at different compression stages and how it depends on the quantization steps used, which is clearly crucial for the final distribution. As a consequence, in realistic situations where a blind analysis is performed and no clue on the quantization steps used is available, it would be hard to quantify the confidence of the outcome, thus limiting the reliability of the method.

- **Chain of processing operations**: when forensic detectors are targeted to the detection of a specific processing, they are rarely robust to changes in the forgery process. This is an intrinsic problem in the forensic analysis, since the traces left in the object are in any case due to a specific operation and may be more or less sensitive to pre- and post-processing, even if their mathematical model is accurate and complete. Indeed, in a realistic scenario, an object most likely undergoes a chain of operators, for instance blurring/compression, or compression/resampling/compression, and so on. The problem of modeling completely the traces left by a cascade of operations is rather demanding. First attempts in this direction have been proposed, addressing certain operations like linear filter/compression [36], compression/resizing/compression [20], compression-contrast enhancement-compression [53]. Moreover, the related problem of multimedia phylogeny [98, 39] is currently intensively investigated and nowadays represents one of the main challenges of multimedia forensics.

- **General adversarial framework**: counter-forensic techniques significantly compromise the performance of traditional forensic methods, but they typically introduce a distortion that can lead to a visual degradation of the image and depends on the strength of the attack. A strongest action guarantees a good recovery of the statistical properties considered but decreases the visual quality; a lightest one preserves the visual features but might be not effective in defeating the forensic tools. With this respect, we can notice that the optimality of anti-forensics techniques is rarely discussed, and a specific forensic detector is usually targeted [12]. Moreover, a strong distortion would make the counterforensic action detectable [131] and useless in practical scenarios.
  Significant steps forward have been done in this direction: in [11] and [33, 34] the transformation of the histogram that minimizes the Mean Squared Error (MSE) distortion is applied, guaranteeing an effective anti-forensics is theoretically derived. In

other words, the optimal attack (in the MSE sense) against histogram-based forensic detectors is obtained.
A promising direction that is currently investigated, is to study the interplay between the forensic analyst and the adversary by means of game theory, as it has been done in [123] and [13]. In this framework, a number of theoretical results on the asymptotic game equilibrium under different hypotheses [14, 15] have been obtained, going toward a general theory in adversarial multimedia forensics.

- **Fusion of the outcomes from different techniques**: in addition to the optimal design of single detectors, another problem arising in the forensic analysis is how to fuse and exploit multiple responses from the forensic tools.
  Indeed, as the current literature offers different methods for facing a wide range of manipulations, an analyst would likely exploit a set of detectors in analyzing an image. This can lead to inconsistencies in the different results and, hence, to uncertainty in the final decision. How to deal with this variety and embed the different results in a fusion system remains an open problem. However, research is quite active in this directions and some approaches have been presented [56, 54, 10], relying on statistical theories. Moreover, attention has been devoted to adversary-aware settings [13, 57].

- **Localization of tampered regions**: most existing techniques for tampering detection are not able to provide automatically the location of the suspect part, or they require the knowledge of a region of interest, to be compared with the external area [56]. Given the relevance of the problem in application scenarios, research in this direction is extremely active and and number of recent approaches show significant advances [9, 41, 142, 7, 60, 19].

2.2 JPEG image forensics

The JPEG compression scheme consists in the steps depicted in Fig.2.1. The core of the compression algorithm is the quantization of 8×8-block DCT coefficients at a certain frequency with a quantization step q, depending on the chosen quality level. Thus, being x the value of a generic DCT coefficient, during the compression it is transformed into

$$x_q = sign(x) \cdot round\left(\frac{|x|}{q}\right) \cdot q,$$

where $sign(\cdot)$ is the sign function and $round(\cdot)$ is the rounding function to the nearest integer. Such quantization, together with the partition in 8×8 blocks, leaves typical artifacts both in the pixel and DCT domain, that are widely studied and used for forensic purposes.

Fig. 2.1 also represent the two possible paths in the digital history of the image, both studied in the literature: it can be decompressed (potentially manipulated) and then re-saved in uncompressed format (TIFF, PNG, BMP, GIF, ecc...); it can be decompressed (potentially manipulated) and recompressed in JPEG format a number of times. Moreover, the presence of a smart counterfeiter might imply a counterforensic phase before the forensic analysis.
In the following, we review the main approaches presented in the literature and identify the more relevant limitations of existing tools to the forensic problems highlighted in red in Fig. 2.1, for which novel contributions are proposed in the following chapters.

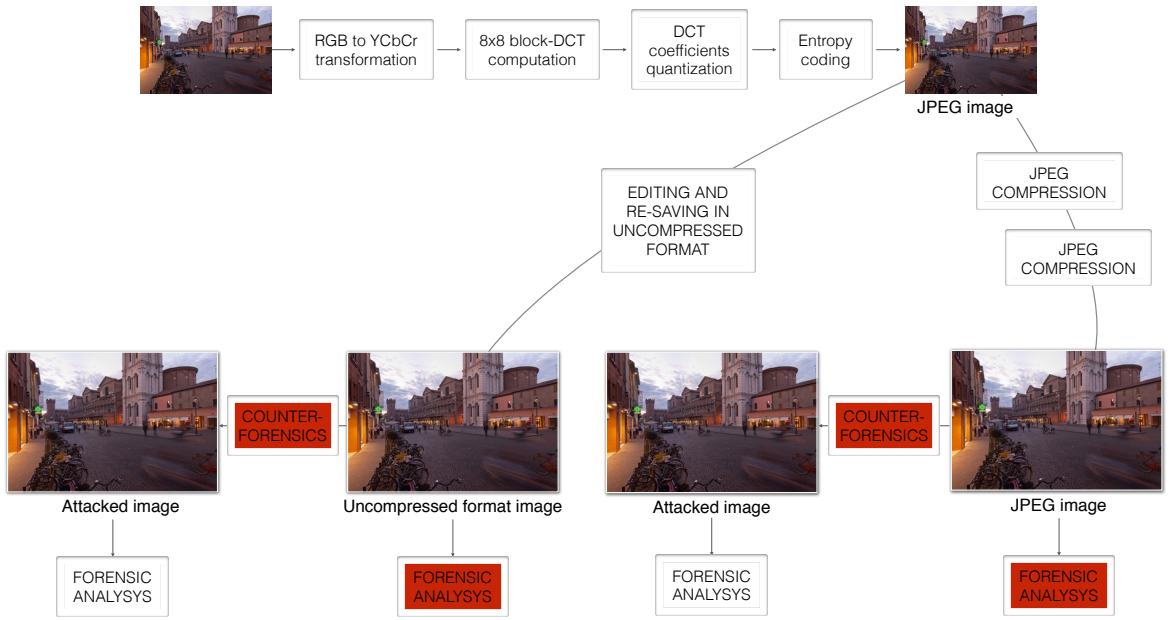


Figure 2.1: Pipeline of the JPEG compression scheme and possible life cycles of compressed images.

2.2.1 Detection of compression traces in uncompressed image formats

In several scenarios, a digital image is available in an uncompressed format (for instance, photographic images delivered in TIFF format), while it potentially underwent prior processing or coding. In these cases, we might be interested in deciding whether that image had been previously compressed and which were the compression parameters being used. One of the first approaches was proposed in [48]: there, the blocking artifacts left by a JPEG compression in the pixel domain are exploited, and a detector based on inter- and intra-block pixel differences is designed. Such values are combined in a final statistic K, expressing the strength of blocking artifacts, and images presenting a value of K higher than a certain threshold are classified as compressed. In the same paper, a procedure based on ML estimation of the used quantization table is proposed. An improved version is presented in [95], where the joint detection of both the quantization table and the used color space transformation is achieved.
As the quantization of the 8×8-block DCT coefficient represents the core of the JPEG compression procedure and leaves characteristic footprints, several methods on JPEG images focus on the analysis of the DCT domain for extracting information on the compression history. In [88], the distribution of DCT coefficients after quantization and reprojection on the pixel domain is studied: in particular, the authors observe how the DCT coefficients behave differently around 0 when the image is pristine or previously compressed. Such different behaviours are captured in a 1D feature, discriminating between original and compressed images; for the latter case, a simple procedure is proposed for estimating the quantization steps.
Both previous methods are characterized by a low complexity and good performance, also in case of small images; on the other hand, the used statistics present a quite different behaviour when varying the size of the image and therefore the performance is strongly dependent on the initial set of images used for determining the optimal threshold.
Another statistic that has been explored in image forensics is the distribution of the First Significant Digits (FSD) of the DCT coefficients. Indeed, when the DCT coefficients are quantized, their FSDs change together with their distribution. In particular, for uncompressed images we have that the FSDs follow a logarithmic distribution, known as Benford's law, which is perturbed when a quantization occurs. Driven by this observation, the authors in [59] proposed a JPEG compression detector based on an SVM classifier which uses as feature the empirical frequencies of the nine FSDs on all the DCT coefficients in the image. The method achieves good results on the considered dataset and requires a relatively low computational complexity; however, it does not provide an estimate of the quality factor or quantization table used, since no theoretical model for the FSD distribution is available, and the results are strongly dependent on the dataset.
Recently, a first approach based of Benford–Fourier coefficients has been proposed in [109], inspiring the design of our novel approaches (see Section 3.1).

MAIN CURRENT LIMITATIONS:

- *a theoretical model for the statistics used is not available, thus generating uncertainty in the case of different parameters*

- *a training phase on a preliminary set of images is needed, being prone to dataset-dependency issues*

2.2.2 Detection of multiple compression traces in JPEG images

Since the JPEG format is adopted in most of the digital cameras and image processing tools, we can expect that a forger will open a JPEG image, modify it and re-compress it a number of times, probably with a different quality factor. Hence, the traces of multiple compression, or inconsistencies of such traces between areas in the same image, can represent a proof of tampering.
In addition, the second compression might not respect the 8×8 grid of the previous one, thus leading to two cases: *aligned* and *non-aligned* double JPEG compression.

For the former one, a great number of methods rely on the analysis of DCT coefficient first-order statistics, i.e., the histogram. An effective approach was proposed in [86], and later improved in [110]. Here, it is observed that consecutive quantizations introduce periodic artifacts and shifted peaks into the histogram of DCT coefficients: this behavior is called *double quantization effect* and represents a significant trace of aligned double JPEG compression (see example in Figure 2.2). Clearly, it is related to the values of the two quantization steps used and, in addition, it helps estimating the primary quality factor. The double quantization effect was also analyzed in detail by authors in [112],

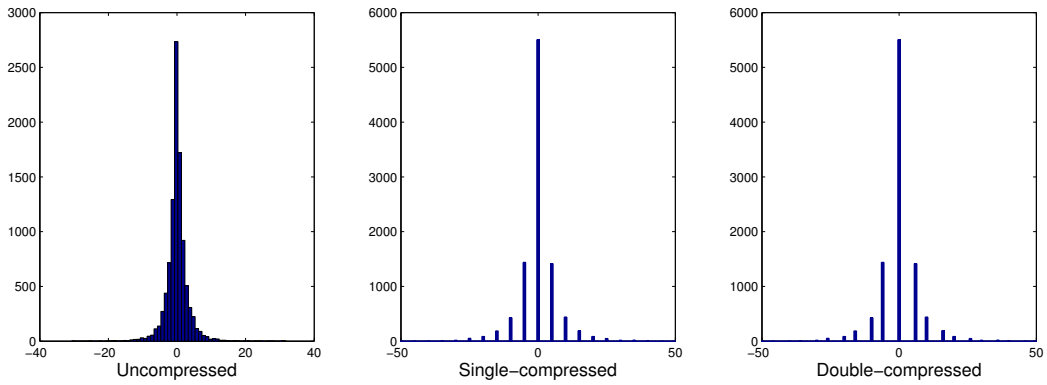


Figure 2.2: Histogram of DCT coefficients at a certain frequency; quantization steps are 5 and 2.

that propose a new statistical model for the artifacts of double compression.
After these two important approaches, a number of refinements based on different analyses of DCT histogram have been proposed, leading to higher accuracy in the forensic classification and in the estimation of the primary quality factor.
In [50], a different perspective is adopted: exploting the idempotency of the coding operation, JPEG images are recompressed with different quality factors and if at least one of them presents a high correlation with the starting one, it was likely previously compressed.
Histogram of FSDs at certain frequencies of the DCT are also exploited for discriminating

single and double compressed images [81], or even images compressed different number of times [92].

Regarding non-aligned double compression, the analysis of blocking artifacts in the pixel domain can once again provide useful information. Indeed, in [89] a Blocking Artifact Characteristics Map (BACM) is computed, starting from the approach in [48]. An asymmetric BACM will reveal the presence of misaligned double compression; some features are then extracted from the BACM and fed to a classifier in order to understand whether a portion of the image underwent non-aligned quantization or not.
A similar method is presented in [30], where the blocking artifacts are measured by means of first-order derivative and fitted to a linear model. Features related to the probability map of each pixel following this model are fed to an SVM.
A different approach is adopted in [18], where the main idea is to detect non-aligned double compression by measuring how DCT coefficients cluster around a given lattice, defined from the JPEG quantization table, for any possible grid shift. In addition, the parameters of the lattice also give an estimation of the primary quantization table. Such method outperforms results obtained with methods based on blocking artifacts and is more robust to quantization parameters. In [17], the same authors present a tampering localization procedure, so that the suspect region does not need to be manually selected.

Finally, few methods aim at detecting either block-aligned or misaligned recompression at the same time have recently been proposed. In [31], a set of features is computed to measure the periodicity of blocking artifacts, perturbed in presence of non-aligned double compression, and another set to measure the periodicity of DCT coefficients, perturbed when an aligned double compression is applied. By combining the two groups, nine features are used to train a classifier, detecting whether an image has undergone double JPEG compression. Moreover, authors in [19] present an improved unified statistical model and an algorithm that aumatically computed a likelihood map, indicating for each 8×8 block the probability of being double compressed.

MAIN CURRENT LIMITATIONS:

- *a lighter primary compression is hardly detected*
- *method typically address double compression*
- *detection of high quality compression is rarely assessed*

2.2.3 JPEG counterforensics and adversarial perspectives

Special attention has been paid to the case of JPEG compression counterforensics. Indeed, the problem of concealing compression artifacts has been faced in [124]. Here, an anti-forensic dither is added to DCT coefficients of a JPEG image, thus restoring the original statistics and destroying the artifacts introduced during the compression. As a result, the image looks uncompressed and, if compressed again, it is classified by forensic detectors as single compressed. Such procedure is tested against different forensic methods and turns out to be very effective for the adversary. This approach has then been enhanced by applying a spatial distribution that minimizes the degradation due to noise addition [131]. With the same intention, a different anti-forensic scheme has been proposed in [47],

where a cost function expressing the distortion is minimized by means of a variational approach.
In response to this kind of attacks, techniques aimed at detecting the anti-forensic action in [124] has been developed [80] [132], based on peculiar effects introduced in the image by that specific algorithm.

MAIN CURRENT LIMITATIONS:

- *optimality of the methods in terms of distortion introduced is rarely addressed*
- *lack of a general adversarial framework*

2.3 Benchmarking

Before moving to the presentation of the novel contributions in the field, we briefly discuss some aspects of the experimental setting we adopted in our validation tests.
First, experiments on JPEG images have generally been performed in Matlab environment, using built-in functions to read and write images. When specified, `libjpeg` and `libtiff` libraries, released by the Independent JPEG Group (IJG) and used in common software, are also used. Throughout the thesis, we refer to the IJG standard quantization tables computed as function of a quality factor $QF \in \{1, 2, \ldots, 100\}$, as done by both the `libjpeg` library and the Matlab built-in encoder.

Moreover, a common benchmarking of algorithm performance is still an open issue in image forensics [44]. Despite the number of general image datasets available online (e.g., MIRFlickr [67]), only few of them are suitable for our purposes due to the uncontrolled nature of the provided images in terms of compression history and kind of processing applied. Indeed, in order to assess the performance of our models, we need images that are uncompressed and not post-processed, as their pristine condition is an essential starting point for building the testing sets needed. Because of that, we selected the following four datasets of uncompressed format images:

- LIU: subset of the database used in [84], composed of 1000 bitmap images (256×256).
- UCID: database proposed in [120], composed of 1338 TIFF images 512×384. Originally proposed for image retrieval validation, it is probably the most used benchmarking dataset in image forensics.
- DRESDEN: database proposed in [62], particularly oriented to camera-based source identification techniques. We used the 1488 images in TIFF format, whose size ranges from 3072×2304 to 4352×3264.
- RAISE: database proposed by our research group in [44] as a benchmarking tool for image forensics. We used subset packages of the original database, composed of 8156 images available both in TIFF and NEF format.

The novel methods developed in this work are tested on all or a subset of such four datasets. This allows us to validate our algorithms on extremely diverse images in terms of content, resolution, acquisition parameters. In particular, the size of the image will be

Figure 2.3: Image size proportions among different datasets.

relevant for the tests in Chapter 4, where size-adaptive detectors are developed. Fig. 2.3 visually represents the different datasets and their size proportions.

Chapter 3

Single compression traces in uncompressed format images

In this chapter, we present the novel solutions proposed for the forensic problem of detecting traces of JPEG compression in images that are stored in uncompressed formats. We first introduce and define the Benford-Fourier coefficients, a mathematical tool that we extensively exploit in our analysis. Then, the statistical characterization of never compressed and previously compressed images is illustrated. Finally, the design of the final detectors is presented, together with results on the benchmarking datasets.

Acknowledgement

I would like to thank Prof. Fernando Pèrez-Gonzàlez for the wise co-supervision of this research, which was partially conducted during my visiting internship within the Signal Processing in Communications Group of the University of Vigo.

Figure 3.1: Forensic scenario considered in this Chapter.

We stressed in Chapter 2 the importance of having a solid theoretical backround behind the design of forensic detectors. For this reason, we tackle from a theoretical perspective the problem of detecting the traces of a previous JPEG compression in images that are stored in uncompressed formats. Such issue appears when the forensic analysis is performed on images supposedly taken by a device set to provide raw images (like professional or semi-professional cameras) or, in general, in every situation where the subject image is supposed to be never compressed, and the presence of JPEG compression traces would suggest that the image has been taken from a different camera or it has been already processed by someone. Indeed, although the JPEG standard represents the most used format for digital images, the need for analyzing uncompressed formats arises, for instance, when professional photographic images are involved. In this case, it is common to deliver images also in uncompressed format (mostly TIFF) to preserve quality.

The closed-form statistical analysis of Benford–Fourier coefficients allows us to define a hypothesis testing framework where the null hypothesis is the pristine condition of the image, and the alternative hypothesis is represented by a previous compression. Here, we propose three novel tests based on different statistical schemes, namely the R-test, $\log \mathbf{L_0}$-test and the λ-test, with the aim of discriminating images that have never compressed from images previously compressed, as depicted in Fig. 3.1.

An interesting peculiarity of the proposed methods is that the statistical description on the BF coefficients, derived analytically, explicitly depends on the number of DCT coefficients considered, i.e., it is related to the size of the subject image. Moreover, all the statistical parameters involved in the model are estimated directly from the data without relying on any predetermined dataset. This results in size-adaptive JPEG compression detectors, which do not require any training phase. Experimental results on several datasets and JPEG compression parameters show the benefits of this approach with respect to state-of-the-art methods.

3.1 Benford–Fourier coefficients

Benford–Fourier coefficients have been originally introduced in [108] and have a precise mathematical meaning which makes them extremely suitable for the forensic problem considered.

For the sake of clarity, in the following we will indicate univariate real or complex random variables with capital letters, whose realization will be represented by the corresponding lower case letters.

Then, let X be a random variable representing the non-zero DCT coefficients and f_X its probability density function; we suppose f_X is symmetric with respect to 0. Then, we define the random variable Z whose values are in $\mathbb{R}_0^+$ such that

$$f_Z(z) = 2 \cdot f_X(z), \qquad \forall z \in \mathbb{R}_0^+$$

By doing so, Z models the behavior of $|X|$ in $\mathbb{R}_0^+$ (f_X is symmetric), i.e. discarding the value 0 as possible outcome. Then, we define the random variables

$$Z' \doteq \log_{10} Z$$

$$\tilde{Z} \doteq \log_{10} Z \mod 1,$$

representing the absolute valued positive DCT coefficients in the logarithmic and modular logarithmic domain, respectively. The r.v. $\tilde{Z}$ is particularly relevant because of its relationship with the pdf of the FSD of X [109] and it has been exploited in image counter-forensic techniques [100] [34].

Now, the *Benford-Fourier (BF) coefficients* in $n \in \mathbb{N}$ are defined as the Fourier transform of $f_{Z'}(z')$ evaluated in $2\pi n$, i.e.,

$$a_n = \int_{-\infty}^{+\infty} f_{Z'}(z') e^{-j2\pi n z'} dz' = \int_{-\infty}^{+\infty} f_Z(z) e^{-j2\pi n \log_{10} z} dz. \tag{3.1}$$

Such coefficients turn out to be particularly suitable for characterizing the DCT coefficient behaviour since they have a key role in the statistical description of $\tilde{Z}$: in fact, in [108] it has been showed that, for a generic continuous r.v. Z, we have

$$f_{\tilde{Z}}(\tilde{z}) = 1 + 2 \sum_{n=1}^{+\infty} |a_n| \cos(2\pi n \tilde{z} + \phi_n), \qquad \tilde{z} \in [0, 1). \tag{3.2}$$

Moreover, in [108] the authors show that if X is a Generalized Gaussian (GG) r.v. with standard deviation σ and shaping factor ν, i.e.,

$$f_X(x) = A e^{-|\beta x|^\nu}, \qquad x \in \mathbb{R},$$

$$\beta = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}}, \qquad A = \frac{\beta \nu}{2\Gamma(1/\nu)},$$

the theoretical expression of a_n, $n \in \mathbb{N}$, and its magnitude can be derived as functions of the GG parameters:

$$\begin{aligned} a_n &= \int_{-\infty}^{+\infty} f_{Z'}(z') e^{-j2\pi n z'} dz' \\ &= \frac{2A}{\beta\nu} e^{j\frac{2\pi n \log\beta}{\log 10}} \Gamma\Big(\frac{-j2\pi n + \log 10}{\nu \log 10}\Big), \end{aligned}$$

$$|a_n|^2 = \prod_{k=0}^{\infty} \left[1 + \frac{(2\pi n)^2}{\log^2 10(\nu k+1)^2}\right]^{-1}. \tag{3.3}$$

As seen in (3.3), the magnitude of the coefficients increases with ν and does not depend on the variance of the GG. In Table 3.1, we report the values of $|a_n|$ computed as in (3.3) (where k ranges from 0 to 10^5) for different values of n and ν, as it is well-known that the DCT coefficients of uncompressed images can be modeled by a Generalized Gaussian r.v. with a shaping factor generally ranging from 0.5 to 1.2 [22]. We can notice that, in particular, when $n \geq 3$, these values are always lower than 10^{-4}.

n	1	2	3	4	5	6	7	8
$\nu = 0.5$	$6.1 \cdot 10^{-3}$	$3.2 \cdot 10^{-6}$	$1.1 \cdot 10^{-9}$	$3.3 \cdot 10^{-13}$	$9.0 \cdot 10^{-17}$	$2.2 \cdot 10^{-20}$	$5.5 \cdot 10^{-24}$	$1.3 \cdot 10^{-27}$
$\nu = 0.75$	$2.7 \cdot 10^{-2}$	$1.6 \cdot 10^{-4}$	$7.4 \cdot 10^{-7}$	$3.1 \cdot 10^{-9}$	$1.2 \cdot 10^{-11}$	$4.8 \cdot 10^{-14}$	$1.8 \cdot 10^{-16}$	$6.7 \cdot 10^{-19}$
$\nu = 1$	$5.6 \cdot 10^{-2}$	$1.1 \cdot 10^{-3}$	$1.8 \cdot 10^{-5}$	$2.9 \cdot 10^{-7}$	$4.6 \cdot 10^{-9}$	$6.9 \cdot 10^{-11}$	$1.0 \cdot 10^{-12}$	$1.5 \cdot 10^{-14}$
$\nu = 1.25$	$8.8 \cdot 10^{-2}$	$3.5 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$	$4.5 \cdot 10^{-6}$	$1.5 \cdot 10^{-7}$	$5.4 \cdot 10^{-9}$	$1.8 \cdot 10^{-10}$	$6.2 \cdot 10^{-12}$

Table 3.1: Magnitude of $|a_n|$ for different values of ν and n.

This represents a useful information in JPEG image forensics and suggests that the behavior of the BF coefficients can be used to characterize uncompressed images. Indeed, a first approach in this direction was proposed in [109], where the BF coefficients from the DCT coefficients of the whole image are estimated by computing the FFT of the empirical distribution of $\tilde{Z}$. Then, the first five coefficients (i.e., $n = 1, \ldots, 5$) are used as feature to train an SVM discriminating between natural uncompressed images and images that underwent a JPEG compression, obtaining promising results.

Although the Benford-Fourier coefficients defined as in (3.1) have a precise meaning given by the expression (3.2), we can extend such definition to the entire real line and, in the following, we will consider as BF coefficient at $\omega \in \mathbb{R}$ the complex number given by

$$a_\omega = \int_{-\infty}^{+\infty} f_{Z'}(z') e^{-j\omega z'} dz' = \int_{-\infty}^{+\infty} f_Z(z) e^{-j\omega \log_{10} z} dz. \tag{3.4}$$

3.2 Statistical analysis of Benford–Fourier coefficients

In light of what has been obtained in [109], we address the problem of discriminating compressed images saved in uncompressed format from images that underwent a JPEG compression. Accordingly, for each DCT frequency we want to quantify the probability that the DCT coefficients have never been quantized or they have been previously quantized with a generic step q. To this aim, we consider the BF coefficients at a fixed DCT frequency and develop a statistical model for each of these two cases. Such models will then be exploited for the design of novel hypothesis tests, where the hypotheses of no compression and compression with a quality factor among a predetermined pool are considered. Whereas the hypothesis testing scheme will be described in detail in Section 4.3, in the following we present the statistical models derived for the BF coefficient of a single DCT frequency.

3.2.1 Uncompressed image model

In order to use BF coefficients for analyzing an image, we need a numerical procedure to estimate such coefficients given the subject image.

By looking at (3.4), we can notice that a_ω is the expected value of the complex random variable $g_\omega(Z) = e^{-j\omega \log_{10} Z}$, whose values lie on the unit circle. Thus, as it is usually done in statistics, we can obtain an estimate of $a_\omega = E\{g_\omega(Z)\}$ by considering the sample mean of $g_\omega(Z)$ provided by the DCT coefficients of the image through the different 8×8 blocks. In other words, if we denote as Z^m the r.v.'s representing the DCT coefficients at the chosen frequency in the m-th block (that we suppose to be independent and identically distributed), we can consider the r.v.

$$\hat{A}_\omega \doteq \frac{\sum_{m=1}^{M} e^{-j\omega \log_{10} Z^m}}{M}, \qquad m = 1, \ldots, M \tag{3.5}$$

and define the estimator of a_ω as the realization of $\hat{A}_\omega$

$$\hat{a}_\omega \doteq \frac{\sum_{m=1}^{M} e^{-j\omega \log_{10} z^m}}{M}, \qquad m = 1, \ldots, M \tag{3.6}$$

where M is the total number of 8×8 blocks in the image.

Although the sample mean is a minimum variance unbiased estimator of the expected value (i.e., $E\{\hat{A}_\omega\} = a_\omega$), we should consider the fact that the actual accuracy of $\hat{a}_\omega$ in the estimation of a_ω depends on the size of the sample considered. For this reason, we are interested in studying the distribution of $\hat{A}_\omega$ as a function of the number of samples M.

To this end, we can observe that $\hat{A}_\omega$ is a sum of M independent and identically distributed random variables $g_\omega(Z^m)$. Then, by applying the Central Limit Theorem (CLT) to the real and imaginary parts of $\hat{A}_\omega$, we have that their distribution is asymptotically

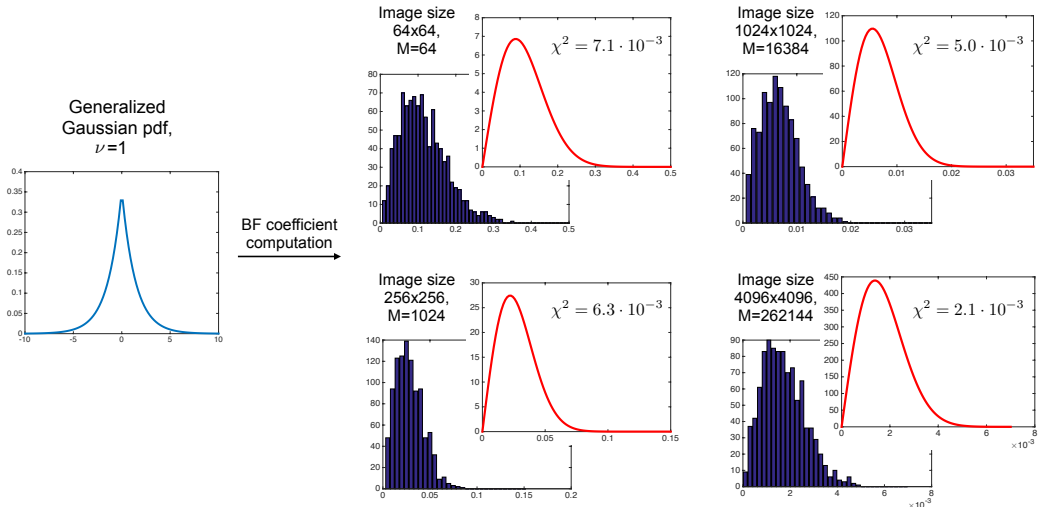


Figure 3.2: The figure depicts the behaviour of BF coefficients when varying the number M of samples used in the sum (4.1). For $M = 64, 1024, 16384, 262144$, we generated 1000 sets of M elements distributed according to a zero-mean Generalized Gaussian pdf with $\nu = 1$ (which is common for DCT coefficients of natural images) and varying standard deviation. Then, the estimate of BF coefficient magnitude $|\hat{a}_\omega|$ (with $\omega = 8\pi$) have been computed on each set and their histograms for the different values of M are plotted (blue bars). The red curves are the Rayleigh pdfs (3.8) determined by the corresponding value of M in each case. The goodness of fit of the proposed model is confirmed by the values of the χ^2 statistics (computed on 10 equally spaced bins from 0 to the highest magnitude value obtained in each case).

Gaussian with expected values $\Re(a_\omega)$ and $\Im(a_\omega)$, respectively [16]. In other words,

$$\hat{A}_\omega = a_\omega + W_0,$$

where W_0 is a zero-mean complex normal random variable.

A necessary and sufficient condition for W_0 to be circularly symmetric (i.e., with real and imaginary parts independent and identically distributed [16]) is that $E\{W_0^2\} = 0$. Starting from the definition of $\hat{A}_\omega$, it is easy to prove that

$$E\{W_0^2\} = E\{(\hat{A}_\omega - a_\omega)^2\} = \frac{1}{M}(a_{2\omega} - a_\omega^2). \tag{3.7}$$

Hence, $|E\{W_0^2\}| \leq (|a_{2\omega}| + |a_\omega^2|)/M$ and, by looking at Table 3.1, we can conclude that the value of (3.7) will be very close to 0 (for instance, when $\nu = 1$ and $\omega = 6\pi$ its order of magnitude is 10^{-11}). Therefore, $\hat{A}_\omega$ is approximately a circular bivariate normal r.v. with non-zero mean.

It is well known that the r.v. $R \doteq |\hat{A}_\omega|$ follows the Rice distribution with mean parameter $|a_\omega|$ and scale parameter σ, where σ is the standard deviation of both its real and imaginary parts [115]. Similarly as before, we can now obtain σ^2 by exploiting the

fact that for a Rice distribution

$$\sigma^2 = \frac{E\{|\hat{A}_\omega|^2\} - |a_\omega|^2}{2} = \frac{1}{2M}(1 - |a_\omega|^2).$$

As we observed, $|a_\omega|$ is lower than 10^{-4} when $\omega \geq 6\pi$ and we can reasonably assume $|a_\omega| \approx 0$, thus considering the special case of Rice distribution with mean parameter 0, i.e., the Rayleigh distribution with scale parameter $\sigma = 1/\sqrt{2M}$. According to this, we can define $p(\hat{a}_\omega|NQ)$ (NQ means "never quantized") as the probability density function of obtaining a BF coefficient $\hat{a}_\omega$ under the hypothesis of no previous quantization and compute it as follows:

$$p(\hat{a}_\omega|NQ) = 2M|\hat{a}_\omega|e^{-M|\hat{a}_\omega|^2}, \tag{3.8}$$

where the expression on the right is the Rayleigh pdf with $\sigma = 1/\sqrt{2M}$. By considering its properties, we have that $|\hat{a}_\omega|$ is in any case an overestimate of $|a_\omega| = 0$, where its mean is given by $\frac{1}{\sqrt{M}} \cdot \frac{\sqrt{\pi}}{2}$ (the expected accuracy increases linearly with $\sqrt{M}$) and its variance is given by $\frac{1}{M} \cdot \frac{4-\pi}{4}$ (the expected accuracy variance decreases linearly with M).

An example of the model is showed in Fig. 3.2.

3.2.2 Compressed image model

When computing the DCT from an image stored in uncompressed format that was previously compressed, the DCT coefficients at a certain frequency have a distribution like in Fig. 3.3a. The error affecting the histogram is due to the quantization in the pixel domain after the block-DCT quantization and the rounding/truncation errors in the DCT computation, and has been modeled in the literature as a Gaussian r.v. [19].

We propose here an alternative statistical description whose accuracy has been assessed

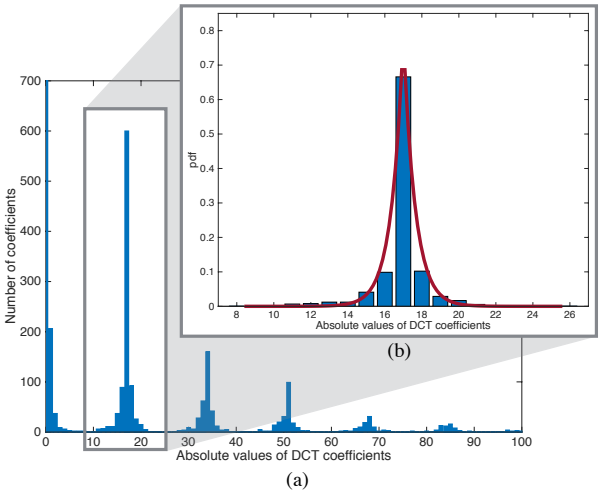


Figure 3.3: In panel (a), histogram of DCT coefficients at a single frequency after quantization with $q = 17$. In panel (b), the coefficients corresponding to the r.v. Z_{17} are reported and the red curve is the Laplacian pdf estimated from the histogram.

by extensive numerical tests. We can restrict our analysis to a single quantization interval $[kq - q/2, kq + q/2[, k \geq 1$ (q is the quantization step) and consider the DCT coefficients contained in such interval. Without any loss of generality, we can suppose $k = 1$ (which is the most natural choice, since the first "bell" in Fig. 3.3a is certainly the one containing more elements and providing more reliable statistics), although the following analysis can easily be extended to any value of k.

If we denote as I_q the interval $[q - q/2, q + q/2[$ and Z_q the r.v. representing the DCT coefficients falling in I_q, we can approximate its distribution with a Laplacian truncated outside the quantization interval, as in Fig. 3.3b. Then, the pdf of Z_q is given by

$$f_{Z_q}(z) = \frac{\mathcal{L}(z; q, \sigma)}{N_{\sigma,q}} \cdot \mathbb{1}_{I_q}(z), \tag{3.9}$$

where $\mathcal{L}(\cdot; q, \sigma)$ is a Laplacian pdf with mean q and standard deviation σ (which is unknown and needs to be estimated), $N_{\sigma,q}$ is the integral of $\mathcal{L}(z; q, \sigma)$ over I_q (so that expression (3.9) is a pdf) and $\mathbb{1}_I(\cdot)$ is the indicator function of I_q

$$\mathbb{1}_{I_q}(z) = \begin{cases} 1 & z \in I_q \\ 0 & z \notin I_q. \end{cases}$$

Starting from this hypothesis, we can define $a_{\omega,q}$ as the Benford-Fourier coefficients of Z_q, and derive its theoretical value as follows:

$$a_{\omega,q} \doteq \int_{-\infty}^{+\infty} f_{Z_q}(z) e^{-j\omega \log_{10} z} dz \tag{3.10}$$

$$= \frac{1}{N_{\sigma,q}} \int_{I_q} \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|z-q|} e^{-j\omega \log_{10} z} dz \tag{3.11}$$

$$= \frac{1}{N_{\sigma,q}\sigma\sqrt{2}} \Big(e^{-\frac{\sqrt{2}}{\sigma}kq} \int_{q-q/2}^{q} e^{\frac{\sqrt{2}}{\sigma}z} z^{-j\frac{\omega}{\ln 10}} dz + e^{\frac{\sqrt{2}}{\sigma}q} \int_{q}^{q+q/2} e^{-\frac{\sqrt{2}}{\sigma}z} z^{-j\frac{\omega}{\ln 10}} dz \Big). \tag{3.12}$$

In other words, assuming a Laplacian distribution of Z_q and given an estimate of σ, we can obtain the theoretical value of $a_{\omega,q}$ from the previous expression in (3.12) by numerically computing the integrals.

Now, we can adopt the same approach as the uncompressed case: consider the sample mean

$$\hat{A}_{\omega,q} = \frac{\sum_{m=1}^{M_q} e^{-j\omega \log_{10} Z_q^m}}{M_q}, \qquad m = 1, \ldots, M_q \tag{3.13}$$

and the estimate of $a_{\omega,q}$

$$\hat{a}_{\omega,q} = \frac{\sum_{m=1}^{M_q} e^{-j\omega \log_{10} z_q^m}}{M_q}, \qquad m = 1, \ldots, M_q \tag{3.14}$$

where M_q is the number of DCT coefficients falling in the interval I_q at the chosen frequency. Then, study its distribution in order to obtain an expression of $p(\hat{a}_{\omega,q}|q)$, the probability of obtaining $\hat{a}_{\omega,q}$ under the hypothesis that the DCT coefficients at the chosen frequency underwent a quantization with step q.

We can partially exploit the logical steps of the uncompressed case reported in Section 3.2.1. Indeed, exactly in the same way, we can conclude that

$$\hat{A}_{\omega,q} = a_{\omega,q} + W_{0,q},$$

where $W_{0,q}$ is a complex zero-mean Gaussian random variable. In addition

$$E\{W_{0,q}^2\} = \frac{1}{M_q}(a_{2\omega,q} - (a_{\omega,q})^2).$$

Differently from the NQ case, we have no clue on the magnitude of $a_{2\omega,q}, a_{\omega,q}$, thus we cannot claim that $E\{W_{0,q}^2\} \approx 0$ and $W_{0,q}$ is circularly symmetric.

Because of that, we consider the distribution of $\hat{A}_{\omega,q}$ in the complex plane and we study the real and imaginary parts of $W_{0,q}$. For the sake of simplicity, we will denote them as W_r and W_i, respectively, (i.e., dropping the dependence on ω and q) and treat their joint pdf $f_{W_r,W_i}(w_r, w_i)$ as a zero-mean real bivariate Gaussian[1], so that

$$p(\hat{a}_{\omega,q}|q) = f_{W_r,W_i}(\Re(\hat{a}_{\omega,q} - a_{\omega,q}), \Im(\hat{a}_{\omega,q} - a_{\omega,q})). \tag{3.15}$$

The analysis is slightly harder than the NQ case, since here we need to determine the parameters of a real bivariate Gaussian whose pdf is given by

$$f_{W_r,W_i}(w_r, w_i) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\Big(-\frac{1}{2} \begin{bmatrix} w_r & w_i \end{bmatrix} \Sigma^{-1} \begin{bmatrix} w_r \\ w_i \end{bmatrix} \Big),$$

$$\Sigma = \begin{bmatrix} \sigma_{W_r}^2 & Cov(W_r, W_i) \\ Cov(W_r, W_i) & \sigma_{W_i}^2 \end{bmatrix}.$$

In other words, we need the three different entries of the covariance matrix, necessary for computing $f_{W_r,W_i}(w_r, w_i)$. All of them have been theoretically derived in order to obtain closed form expressions and are reported in the following:

- $\sigma^2_{\mathbf{W_r}}, \sigma^2_{\mathbf{W_i}}$**:**

[1] In particular, we treat $W_{0,q}$ as a real bivariate r.v. instead of a complex normal r.v., i.e., in terms of variance and covariance of the two single parts instead of complex covariance and pseudo-covariance (as it is usually done when dealing with complex r.v.'s).

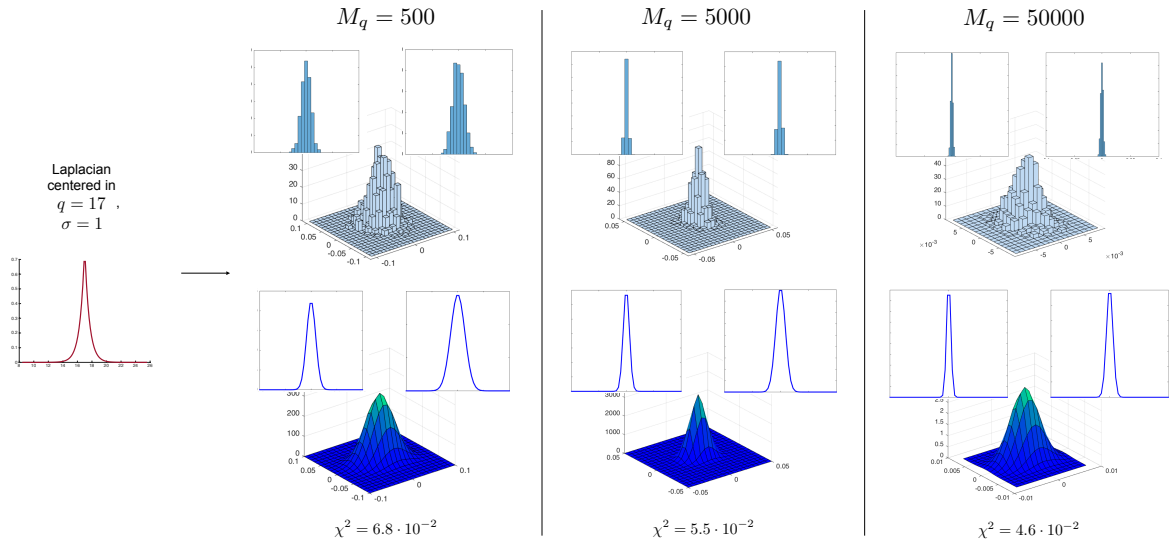


Figure 3.4: The figure depicts the behaviour of BF coefficients in case of compression when varying the number of samples belonging to the interval I_q considered. We considered the case of $q = 17$ and, for $M = 500, 5000, 50000$, we generated 1000 sets of M elements distributed according to a Laplacian pdf with mean 17 and standard deviation $\sigma = 1$. Then, the estimate of the complex BF coefficient $\hat{a}_{\omega,17}$ (with $\omega = 8\pi$) have been computed on each set and its histograms (for the complex values, the real and imaginary parts) for the different values of M_q are plotted (pale blue bars). The blue curves beside are the theoretical pdf for the complex r.v. and the marginal pdfs of the real and imaginary parts, determined as in Section 3.2.2. The match between the histograms obtained and the pdfs derived theoretically is confirmed by the value of the χ^2 statistics, computed on 25 two-dimensional bins (5 along the real part and 5 along the imaginary part).

We have that $\sigma^2_{W_r}$ and $\sigma^2_{W_i}$ are given by the variances of the r.v.'s $\Re(e^{-j\omega \log_{10} Z_q})$ and $\Im(e^{-j\omega \log_{10} Z_q})$, respectively, divided by the number of summands in the sample mean (3.14). The exact expressions are:

$$\sigma^2_{W_r} = \frac{1}{M_q} \int_{-1}^{1} \sum_{z'' \in D_c} \frac{e^{-\frac{\sqrt{2}}{\sigma}|10^{\frac{z''}{\omega}} - q|} 10^{\frac{z''}{\omega}}}{\sigma\sqrt{2} N_{\sigma,q} \omega} \frac{c^2 \ln 10}{\sqrt{1-c^2}} dc - \Re(a_{\omega,q}) \tag{3.16}$$

$$\sigma^2_{W_i} = \frac{1}{M_q} \int_{-1}^{1} \sum_{z'' \in D_s} \frac{e^{-\frac{\sqrt{2}}{\sigma}|10^{\frac{z''}{\omega}} - q|} 10^{\frac{z''}{\omega}}}{\sigma\sqrt{2} N_{\sigma,q} \omega} \frac{s^2 \ln 10}{\sqrt{1-s^2}} ds - \Im(a_{\omega,q}) \tag{3.17}$$

where D_c and D_s are discrete finite set of points that can be determined from the values of c and s. A complete derivation can be found in the Appendix A.

Note that an estimate of σ (the parameter of the Laplacian) is necessary and can be obtained from the data by means of an unbiased sample variance.

- **Cov($\mathbf{W_r}, \mathbf{W_i}$):**

We have that

$$E\{W_{0,q}\} = E\{W_r^2\} - E\{W_i^2\} + 2jE\{W_r W_i\} \tag{3.18}$$

$$= \frac{a_{2\omega,q} - (a_{\omega,q})^2}{M_q} \tag{3.19}$$

and $Cov(W_r, W_i) = E\{W_r W_i\}$. Then, we can obtain

$$Cov(W_r, W_i) = \frac{\Im\left(\frac{1}{M_q}(a_{2\omega,q} - (a_{\omega,q})^2)\right)}{2} \tag{3.20}$$

Fig. 3.4 depicts an example of the pdfs obtained by fixing a Laplacian distribution and generating sample vectors with varying length M_q. It can be noticed that derived statistical models fit the data very accurately.

Finally, we can summarize the necessary steps to obtain $p(\hat{a}_{\omega,q}|q)$ as follows:

- identify the set of DCT coefficients falling in I_q,
- estimate the parameter σ of the Laplacian distribution by means of an unbiased sample variance,
- compute the theoretical value of $a_{\omega,q}$ by means of (3.12) (in this phase numerical integration will be used),
- compute $\sigma^2_{W_r}$, $\sigma^2_{W_i}$ and $Cov(W_r, W_i)$, by means of (3.16), (3.17) and (3.20),
- compute the estimate $\hat{a}_{\omega,q}$ from the DCT coefficients as in (3.14) obtain its probability under the hypothesis of quantization with step q as in (3.15).

3.3 Hypothesis tests

Given the statistical characterization of BF coefficients under both the hypotheses of no previous quantization and quantization with a generic step, we can now exploit such statistical descriptions for JPEG compression detection.

The statistical models derived allow us to formulate different hypothesis tests and use different discriminatory statistics. In particular, the null hypothesis is always given by

$$\mathbf{H_0}\text{: the image has never been compressed}$$

and we will differentiate the tests according to the alternative hypothesis $\mathbf{H_1}$ considered (simple or composite) and the set F of DCT frequencies considered in the analysis. For the sake of clarity, we will indicate the BF coefficients at a certain DCT frequency $f \in F$ as $\hat{a}_\omega^f$.

In the following, we propose three tests involving different amount of information from the image under investigation, together with experimental validation on the benchmarking datasets.

3.3.1 Single-frequency simple alternative hypothesis: the R-test

In this first test, the alternative hypothesis is given by the fact that the image underwent a JPEG compression. Thus, $\mathbf{H_0}$ is as in 3.3, while

$$\mathbf{H_1}\text{: the image has been previously compressed}$$

Moreover, the DCT frequency considered is only one:

$$F = \{f\}, \quad f \in \{1, \dots, 64\}$$

Then, the statistics used is the magnitude of the BF coefficients $R = |\hat{A}_\omega^f|$ at a certain DCT frequency f and some $\omega \geq 6\pi$. We know from Section 3.2.1 that the probability density function of obtaining a BF coefficient $\hat{a}_\omega$ under the hypothesis of no previous quantization is the following:

$$p(\hat{a}_\omega|NQ) = 2M|\hat{a}_\omega|e^{-M|\hat{a}_\omega|^2}. \tag{3.21}$$

Then, we can design a test with an upper threshold, that can be derived by fixing a significance level and using the cdf of the Rayleigh distribution in (3.21), given by

$$F_R(r) = 1 - e^{-Mr^2}.$$

As a general approach, once a significance level α is fixed, we can reject the null hypothesis when the value of r obtained from the image is such that $r \geq \tau_\alpha$ where τ_α is computed such that

$$1 - F_R(\tau_\alpha) = \alpha.$$

3.3.2 Multiple-frequency simple alternative hypothesis: the $\log \mathbf{L_0}$-test

The null and alternative hypothesis are the same as before, but in this case we consider a set F of DCT coefficients with cardinality $|F| > 1$.

Thus a number $|F|$ of BF coefficients are available $\hat{a}_\omega^f$ and their value can be combined to obtain a likelihood function value. Indeed, assuming statistical independence between DCT frequencies [19]) we can compute the likelihood function value for the null hypothesis from the probabilities $p(\hat{a}_\omega^f|NQ)$ as

$$\ell_{\mathbf{0}} = \prod_{f \in F} p(\hat{a}_\omega^f|NQ). \tag{3.22}$$

Such value is then thresholded in order to classify an image as never compressed or compressed, by means of a full statistical characterization. Indeed, we can consider the likelihood function itself as a r.v. $\mathbf{L_0}$ depending on the r.v.'s $\hat{A}_\omega^f$ and study its distribution. Thus, we can reformulate expression (3.27) as

$$\mathbf{L_0} = \prod_{f \in F} p(\hat{A}_\omega^f|NQ) \tag{3.23}$$

$$= \prod_{f \in F} 2M|\hat{A}_\omega^f| \exp(-M|\hat{A}_\omega^f|^2) \tag{3.24}$$

where M is the number of DCT blocks in the image.

Equivalently, we can consider its natural logarithm:

$$\log \mathbf{L_0} = |F| \log(2M) + \sum_{f \in F} \log(|\hat{A}_\omega^f|) - M \sum_{f \in F} |\hat{A}_\omega^f|^2 \tag{3.25}$$

$$= |F| \ln(2M) + \sum_{f \in F} \underbrace{\left[\underbrace{\log|\hat{A}_\omega^f|}_{L_f} \underbrace{-M|\hat{A}_\omega^f|^2}_{B_f} \right]}_{S_f} \tag{3.26}$$

where $|F|$ is the cardinality of F.

The r.v.'s $|\hat{A}_\omega^f|$ are i.i.d. (assuming independency among DCT frequencies) and they follow a Rayleigh distribution with scale parameter $\sigma = 1/\sqrt{2M}$. Starting from this we can study L_f, B_f and S_f.

- Each L_f is a log-Rayleigh random variable. According to [116], we have that

$$E\{L_f\} = -\frac{\log M}{2} - \frac{\gamma}{2}$$

$$Var\{L_f\} = \frac{\pi^2}{24},$$

  where γ is the Euler-Mascheroni constant.

- Each B_f is a squared Rayleigh r.v. multiplied by a constant term $-M$. It can be shown (via r.v. transformation) that a squared Rayleigh r.v. with scale parameter σ has an exponential distribution with rate parameter $1/2\sigma^2$ (in our case M). By scaling with a factor $-M$, we have that

$$E\{B_f\} = -1$$
$$Var\{B_f\} = 1.$$

- Each S_f's is sum of two r.v., hence we have that

$$\begin{aligned} E\{S_f\} &= E\{L_f\} + E\{B_f\} \\ &= -\frac{\log M}{2} - \frac{\gamma}{2} - 1 \end{aligned}$$

$$\begin{aligned} Var\{S_f\} &= Var\{L_f\} + Var\{B_f\} + 2Cov\{L_f, B_f\} \\ &= \frac{\pi^2}{24}, \end{aligned}$$

  as the value of the covariance has been derived by means of symbolic computation and is equal to $-1/2$.

- Finally, $\log \mathbf{L_0}$ is a sum of the iid r.v.'s S_f plus a constant term $|F| \log(2M)$. Then, we have that

$$E\{\log \mathbf{L_0}\} = |F| \ln(2M) - |F| \left(\frac{\log M}{2} + \frac{\gamma}{2} + 1 \right),$$

$$Var\{\log \mathbf{L_0}\} = |F| \cdot \frac{\pi^2}{24}.$$

We can notice that the statistical models is properly scaled according to M, the number of DCT coefficients involved in the estimation of the Benford-Fourier coefficients, thus allowing flexibility when analyzing images of different size. Thanks to these results is possible to design a threshold-based test on the value of the likelihood function by exploiting the Chebyshev's inequality:

$$p\left(|\log \mathbf{L_0} - E\{\log \mathbf{L_0}\}| \geq k\sqrt{Var\{\log \mathbf{L_0}\}} \right) \leq \frac{1}{k^2}, k \in \mathbb{Z}.$$

Thus, we can fix a significance level α and by setting $k = \pm\sqrt{1/\alpha}$ we have that the probability that $\log \mathbf{l_0}$ deviates from $E\{\log \mathbf{L_0}\}$ more than k times the standard deviation of $\log \mathbf{L_0}$ is lower than α. By considering that $\log \mathbf{L_0}$ expresses the likelihood of the null hypothesis, we can define the threshold

$$\tau_\alpha = E\{\log \mathbf{L_0}\} - \sqrt{\frac{1}{\alpha}} \cdot \sqrt{Var\{\log \mathbf{L_0}\}},$$

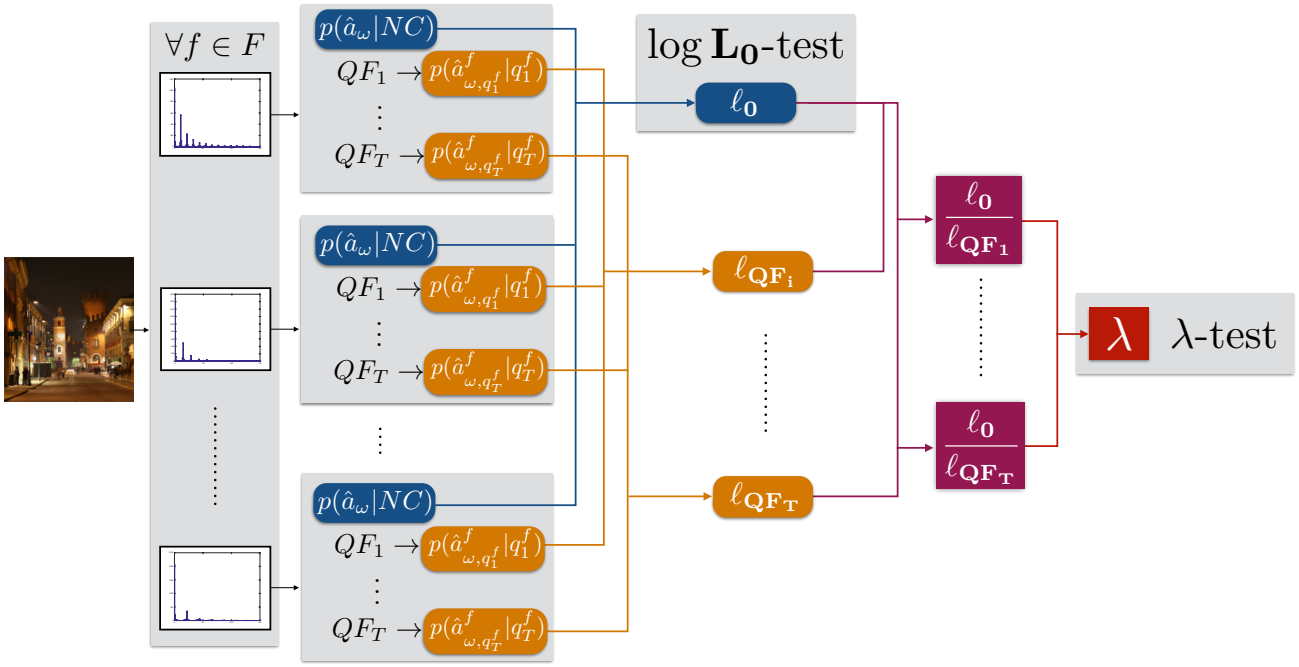


Figure 3.5: Scheme of the proposed JPEG compression detection algorithm.

and design a threshold-based test as follows:

$$\begin{array}{ll} \mathbf{H_0} \text{ is accepted} & \text{if } \log \ell_{\mathbf{0}} \geq \tau_\alpha \\ \mathbf{H_0} \text{ is rejected} & \text{if } \log \ell_{\mathbf{0}} < \tau_\alpha. \end{array}$$

It is worth pointing out that τ_α can be theoretically determined, thus avoiding again the need of any preliminary training on data.

3.3.3 Multiple-frequency composite alternative hypothesis: the λ-test

In this case, we consider a pool of possible quality factors $\{QF_1, \ldots, QF_T\}$ and corresponding quantization tables, and define the null alternative hypotheses as:

$$\begin{array}{c} \mathbf{H_1}\text{: the image has been compressed} \\ \text{with a quality factor among } QF_1, \ldots, QF_T \end{array}$$

Thus, we can consider the alternative hypothesis as composite since it includes a set of possible parameters, the potential quality factors. Now, we propose a discriminative test encompassing statistical models of both null and alternative hypotheses. Also in this case, we propose to automatically determine an acceptance region for the null hypothesis $\mathbf{H_0}$ without the need of any training phase, and provide a final binary output on the subject image (i.e., never compressed or compressed).

The test consists in the application of the Generalized Likelihood Ratio Test (GLRT) [21] to the given problem and its workflow is represented in Fig 3.5. As in 3.25, we compute the BF coefficients $\hat{a}^f_\omega$ for the chosen set F of DCT frequencies and obtain the

likelihood function value for the null hypothesis as

$$\ell_{\mathbf{0}} = \prod_{f \in F} p(\hat{a}^f_{\omega} | NQ). \tag{3.27}$$

Moreover, we want to provide a likelihood value for each parameter of the composite alternative hypothesis, hence for each quality factor considered. To this end, for each frequency f and quality factor QF_i, the corresponding quantization step q_i^f at that DCT frequency is retrieved. Similarly as the null hypothesis, we can compute the BF coefficient estimate $\hat{a}^f_{\omega,q_i^f}$ and its probability value $p(\hat{a}^f_{\omega,q_i^f}|q_i^f)$ can be obtained as in Section 3.2.2, thus obtaining a likelihood function value for each QF_i:

$$\ell_{\mathbf{QF_i}} = \prod_{f \in F} p(\hat{a}^f_{\omega,q_i^f} | q_i^f). \tag{3.28}$$

Finally, according to the GLRT design, the discriminative statistic is given by

$$\lambda = \frac{\ell_{\mathbf{0}}}{\max_{i \in \{1,\dots,T\}} \ell_{\mathbf{QF_i}}}. \tag{3.29}$$

Then, we propose to obtain the final decision on the image as follows:

$$\begin{array}{ll} \mathbf{H_0} \text{ is accepted} & \text{if } \lambda \geq 1 \\ \mathbf{H_0} \text{ is rejected} & \text{if } \lambda < 1. \end{array}$$

Indeed, setting the threshold to 1 means that we reject the null hypothesis as soon as we find an alternative one which achieves an higher value of the likelihood function. It is worth pointing out that this choice is not driven by a theoretical model, as it happens in the two previous tests. Indeed, as a distribution model of λ would be required in order to fix an optimal threshold at a certain false alarm probability upperbound. However, we will see in the next section that the results achieved in the experimental settings considered are comparable to the ones obtained with an optimal threshold, numerically derived from the data by means of a false alarm upperbound criterium.

3.4 Experimental results

We performed extensive tests in order to assess the effectiveness of the proposed approach, both in classifying images as uncompressed or compressed and in identifying portions of the same image showing different compression histories.

In the following, we describe the experimental settings considered and the results obtained on images coming from diverse sources, comparing the performance with existing forensic methods.

3.4.1 Uncompressed vs Compressed discrimination

We first consider images belonging to the four benchmarking datasets (the 2K subset of the RAISE dataset has been used [44]) and apply the three tests to the full-size images.

Other four state-of-the-art methods have been considered for comparisons, in particular the techniques proposed in [109], [59], [48] and [88], denoted as `BF FFT`, `FSD`, `BLOCK` and `DCT`, respectively; all of them have been briefly described in Chapter 2 and require a preliminary training phase (`BF FFT` and `FSD` are SVM-based, while the `BLOCK` and `DCT` methods are threshold-based). Therefore, each dataset was randomly divided into two equal parts, one for training and another one for testing. All the images have been compressed at quality factors $\{90, 80, 70, 60, 50\}$, features were extracted from the ones in the training set and used for obtaining the SVM models or determining the optimal threshold. In this phase, SVM training was performed as suggested in the original papers and the thresholds have been chosen by maximizing the detection rate while keeping the false alarm rate below 1% (where the negatives are represented by uncompressed original images and the positives are the five compressed versions of each image).

All the images were processed as in Sections 3.3.1, 3.3.2 and 3.3.3, thus obtaining a value of R, $\log \mathbf{L_0}$ and λ for each image; we fixed the value of $\omega = 8\pi$ and we limited the analysis to the first 10 AC DCT frequencies in zig-zag order, in order to avoid high number of null coefficients. For each method, we first obtain an optimal threshold from the training set, with same procedure used for the method [48] and [88]. We indicate such thresholds as τ^{R}_{opt}, $\tau^{\log \mathbf{L_0}}_{opt}$ and τ^{λ}_{opt}, respectively. On the other hand, the automatic thresholds for the R- and $\log \mathbf{L_0}$-test are determined as described in Sections 3.3.1 and 3.3.2 by fixing the significance level $\alpha = 0.01$ and are indicated as $\tau^{R}_{0.01}$ and $\tau^{\log \mathbf{L_0}}_{0.01}$, respectively. For the λ-test, $\tau^{\lambda} = 1$.

Then, we consider the images in the testing set, analyzing each of them in their uncompressed version or compressed with quality factor in $\{90, 80, 70, 60, 50\}$, comparing the use of trained and automatic discriminative thresholds.

In Tables 3.2 and 3.3, we report the results for the different methods applied to the test set of every dataset, where the quality factors considered are arranged column-wise. For each method, the false positive rate (**FP**), the true positive rate (**TP**) and the total accuracy (**ACC**) are specified. In each row block the test and the threshold used is specified. The last four row blocks refer to the state-of-the-art methods applied with the SVM models and thresholds derived from the training set. The four datasets are treated separately, i.e., each image is analyzed by using the models/thresholds obtained from the

Table 3.2: Accuracy on the binary classification uncompressed vs compressed (LIU and UCID).

LIU

		90	80	70	60	50
R-test, τ^R_{opt}	**FP**			1.0		
	TP	88.4	95.6	95.0	91.6	89.4
	ACC	93.7	93.7	97.0	95.3	94.2
R-test, $\tau^R_{0.01}$	**FP**			0.6		
	TP	86.2	94.6	93.8	91.0	88.4
	ACC	92.8	97.0	96.6	95.2	93.9
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{opt}$	**FP**			0.4		
	TP	89.8	99.4	99.2	99.4	99.4
	ACC	94.7	99.5	99.4	99.5	99.5
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{0.01}$	**FP**			0.0		
	TP	78.8	97.8	98.8	99.0	99.0
	ACC	89.4	98.9	99.4	99.5	99.5
λ-test, τ^{λ}_{opt}	**FP**			1.2		
	TP	78.2	97.8	97.8	99.4	98.2
	ACC	88.5	98.3	98.3	99.1	98.5
λ-test, τ^{λ}	**FP**			2.8		
	TP	97.2	98.6	99.2	99.8	99.6
	ACC	97.2	97.9	98.2	98.5	98.4
BF FFT	**FP**			25.8		
	TP	99.4	99.8	96.6	98.8	98.2
	ACC	86.8	87.0	85.4	86.5	86.2
FSD	**FP**			7.4		
	TP	87.4	97.8	94.8	87.8	82.8
	ACC	90.2	95.2	92.4	86.1	84.2
BLOCK	**FP**			1.0		
	TP	59.0	77.2	86.4	91.2	93.0
	ACC	79.0	88.1	92.7	95.1	96.0
DCT	**FP**			0.4		
	TP	99.0	98.0	96.4	95.8	94.6
	ACC	99.3	98.8	98.0	97.7	97.1

UCID

		90	80	70	60	50
R-test, τ^R_{opt}	**FP**			1.2		
	TP	99.4	99.7	99.7	99.4	98.5
	ACC	99.1	99.3	99.3	99.1	98.7
R-test, $\tau^R_{0.01}$	**FP**			1.1		
	TP	99.5	99.8	99.8	99.5	98.6
	ACC	99.1	99.3	99.3	99.1	98.7
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{opt}$	**FP**			0.4		
	TP	99.5	99.8	99.8	99.7	100
	ACC	99.5	99.7	99.7	99.6	99.8
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{0.01}$	**FP**			0.0		
	TP	99.0	99.4	99.7	99.7	100
	ACC	99.5	99.7	99.9	99.9	100
λ-test, τ^{λ}_{opt}	**FP**			0.4		
	TP	99.3	99.4	99.9	99.9	99.9
	ACC	99.4	99.5	99.7	99.7	99.7
λ-test, τ^{λ}	**FP**			1.2		
	TP	99.3	99.4	99.7	99.8	100
	ACC	99.0	99.1	99.3	99.6	99.4
BF FFT	**FP**			6.4		
	TP	99.5	99.1	98.0	98.9	98.2
	ACC	96.5	96.3	95.8	96.2	95.8
FSD	**FP**			2.5		
	TP	96.4	99.1	98.6	96.4	95.5
	ACC	97.3	98.2	97.5	96.9	96.5
BLOCK	**FP**			1.3		
	TP	97.3	100	99.8	100	100
	ACC	97.9	99.3	99.2	99.3	99.3
DCT	**FP**			0.4		
	TP	98.3	97.7	97.4	97.4	97.4
	ACC	98.9	98.6	98.5	98.5	98.5

training phase of the same dataset.

We can notice that the proposed approaches achieves good performance in all the datasets. Moreover, we can observe only a slight improvement in terms of accuracy when the optimal thresholds are used instead of the automatic ones. Furthermore, the benefits of the size-adaptive scheme compared to the other methods are particularly clear when their training settings are modified. In Table 7.4, we report the results obtained by replicating the previous experiments but applying the threshold/SVM model obtained from the training phase of a certain dataset to the testing set of a different dataset, and we can see that the performance generally drops. On the other hand, the R, $\log \mathbf{L_0}$ and λ statistics prove to be more stable through the different datasets, as the threshold 1 provides good results in every case.

It is interesting to notice that, differently from the $R-$ and λ-test, the $\log \mathbf{L_0}$ test generally proves to be more accurate on the smaller datasets rather than the bigger ones; for instance, we can notice a higher false alarm rate when the bigger images in the RAISE2K dataset are analyzed. This suggests the use of the $\log \mathbf{L_0}$-test in smaller images or local parts of bigger pictures.

Table 3.3: Accuracy on the binary classification uncompressed vs compressed (DRESDEN and RAISE2K).

		DRESDEN					RAISE2K					
		90	**80**	**70**	**60**	**50**		**90**	**80**	**70**	**60**	**50**
R-test, τ^R_{opt}	**FP**			0.9			**FP**			0.0		
	TP	100	99.9	100	100	99.6	**TP**	100	100	100	100	100
	ACC	99.5	99.4	99.5	99.5	99.3	**ACC**	100	100	100	100	100
R-test, $\tau^R_{0.01}$	**FP**			1.4			**FP**			0.02		
	TP	100	99.9	100	100	99.6	**TP**	100	100	100	100	100
	ACC	99.3	99.2	99.3	99.3	99.0	**ACC**	100	100	100	100	100
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{opt}$	**FP**			0.1			**FP**			0.3		
	TP	100	100	100	100	100	**TP**	100	100	100	100	99.9
	ACC	99.9	99.9	99.9	99.9	99.9	**ACC**	99.8	99.8	99.8	99.8	99.9
$\log \mathbf{L_0}$-test, $\tau^{\log \mathbf{L_0}}_{0.01}$	**FP**			0.6			**FP**			4.3		
	TP	100	100	100	100	100	**TP**	100	100	100	100	100
	ACC	99.7	99.7	99.7	99.7	99.7	**ACC**	97.9	97.9	97.9	97.9	97.9
λ-test, τ^λ_{opt}	**FP**			0.2			**FP**			0.0		
	TP	100	100	100	100	100	**TP**	99.7	99.9	100	100	100
	ACC	99.9	99.9	99.9	99.9	99.9	**ACC**	99.8	99.9	99.9	99.9	99.9
λ-test, τ^λ	**FP**			0.2			**FP**			0.0		
	TP	100	100	100	100	100	**TP**	99.4	99.9	99.9	100	100
	ACC	99.9	99.9	99.9	99.9	99.9	**ACC**	99.7	99.9	99.9	100	100
BF FFT	**FP**			4.0			**FP**			4.7		
	TP	100	99.7	97.3	99.3	99.8	**TP**	99.8	99.8	99.5	98.9	99.6
	ACC	97.9	97.8	96.6	97.6	97.9	**ACC**	97.5	97.5	97.5	97.1	97.4
FSD	**FP**			2.5			**FP**			0.8		
	TP	98.3	100	98.5	91.1	89.1	**TP**	86.3	96.8	98.5	94.4	91.4
	ACC	99.1	99.9	97.9	95.2	94.5	**ACC**	93.0	98.0	98.7	97.0	95.6
BLOCK	**FP**			0.5			**FP**			1.8		
	TP	100	100	100	100	100	**TP**	100	100	100	100	100
	ACC	99.7	99.7	99.7	99.7	99.7	**ACC**	99.1	99.1	99.1	99.1	99.1
DCT	**FP**			0.0			**FP**			0.7		
	TP	99.7	99.8	99.8	99.8	99.8	**TP**	100	100	99.9	99.9	99.9
	ACC	99.9	99.9	99.9	99.9	99.9	**ACC**	99.6	99.6	99.6	99.6	99.6

Table 3.4: Accuracy with training and testing images of different datasets

		90	**80**	**70**	**60**	**50**
		TR: UCID. TS: LIU.				
BLOCK	**FP**			65.9		
	TP	99.7	100	100	100	100
	ACC	66.9	67.0	67.0	67.0	67.0
		TR: LIU. TS: RAISE2K.				
DCT	**FP**			21.8		
	TP	100	100	100	100	100
	ACC	89.1	89.1	89.1	89.1	89.1
		TR: DRESDEN. TS: LIU.				
FSD	**FP**			64.0		
	TP	91.5	90.2	98.0	99.8	98.6
	ACC	63.1	63.1	67	67.9	67.3
		TR: RAISE2K. TS: UCID.				
BF FFT	**FP**			11.8		
	TP	98.7	99.1	98.2	99.5	99.6
	ACC	93.5	93.7	93.2	93.8	93.9

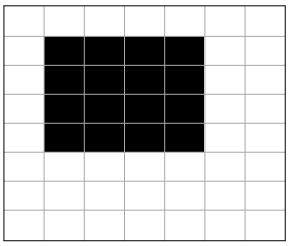
3.4.2 Localization of forged areas

Although the results presented so far were obtained on full frame images, the proposed method can also be applied locally and reveal inconsistencies within the same picture,

thus suggesting that a manipulation occurred. This happens, for instance, when a JPEG image is manipulated by adding one or more parts either coming from uncompressed images or processed so that compression traces disappear. If the image is finally saved in uncompressed format, the original part will present traces of compression while the added part will behave as uncompressed in terms of DCT statistics.

In order to synthetically reproduce such situation, we considered images from the Dresden dataset and compressed them with a random quality factor from 50 to 100. Then, we replaced a square part of 1600×2240 pixels with its original uncompressed version, so that the compression mask is as in Fig. 3.6.

In light of the results obtained in Section 3.4.1, for this localization task we used the $\log \mathbf{L_0}$ test, which proved to be more suitable for smaller pictures and also have a lower computational complexity with respect to the λ-test. In particular, we performed a local analysis by applying the test to non-overlapping windows of 400×560 pixels and by classifying each of them as compressed or non compressed according to a threshold on $\log \mathbf{L_0}$, which is computed as function of the number of DCT blocks in each window (3500 in this case) and the desired significance level (0.01). The results, reported in Fig. 3.6, show that the proposed method achieves very good accuracies also in a local analysis of high resolution pictures, also thanks to its size-adaptive nature.



	Average	Total
FP	0.33	0.36
TP	98.7	98.8
ACC	99.2	99.2

Figure 3.6: Tampering mask and window grid (black and white corresponds to uncompressed and compressed, respectively). The table beside reports in the first column the classification results obtained in the single images averaged on the number of images considered, while the second column refers to the total number of windows classified through all the images.

As further demonstration of the effectiveness of our approach also in non-synthetic settings, we report in Fig. 3.7 an example obtained on a visually compelling manipulated image. In this case, an object coming from a different picture has been inserted into an existing JPEG image and the composite image has been saved in an uncompressed format. We can see from the map that the background is characterized by lower values of $\log \mathbf{L_0}$, which are correctly classified as compressed according to the computed threshold.

3.5 Discussion

We have proposed a novel statistical analysis of BF coefficients, which is exploited to design threshold-based forensic detectors able to reveal the traces of JPEG compression in digital images stored in uncompressed formats. Thanks to the statistical models developed, for all the detectors it is possible to automatically determine the threshold to be

Figure 3.7: Local analysis on a forged image. Overlapping windows of 240×240 pixels have been used.

used, thus avoiding any preliminary training phase on data. The experiments performed on real images of different size and source cameras show that such an approach generally overcomes existing methods and introduces significant benefits in terms of performance. Indeed, both the detectors achieve a very good accuracy through the different kinds of images, thus overcoming dataset-dependency issues, and also provide a valuable tool for the localization of forged areas in certain forensic scenarios.

However, a number of aspects deserve a further analysis and represent future research directions, like the extension of our methods to other compression standards and its evaluation in an adversarial perspective. Moreover, the encouraging results we obtained confirm the effectiveness of BF coefficients in characterizing DCT coefficients and introduces the following chapter, where BF coefficients are used to analyzed images stored in JPEG format.

Chapter 4

Multiple compression traces in JPEG format images

The goal of this chapter is the analysis of digital images in JPEG format, with the aim of determining whether a number of aligned previous compressions occurred. To this end, Benford–Fourier coefficients are again used to study the DCT coefficients of multiple compressed images. A general hypothesis testing scheme is developed, allowing for the reconstruction of the compression chain of an image. The method has been tested in forensic scenarios, where up to three JPEG compressions are considered. Moreover, the performance of the method are evaluated with respect to different ranges of quality factors used and different encoding libraries.

Acknowledgement

This research has been conducted under the co-supervision of Prof. Fernando Pèrez-Gonzàlez.

4.1 Background

In our forensic scenario, the image under investigation is a JPEG file, from which we can extract the DCT coefficients and the quantization table. According to what we assumed in Section 2.3, we can identify the quantization table with a quality factor that we denote as QF_c (*current quality factor*).
In the following, we will build hypotheses on the processing history of the image and each hypothesis will be associated to a certain *compression chain*. A generic compression chain is denoted as $\mathbf{QF} = [QF_1, \dots, QF_L]$, where assuming $\mathbf{QF}$ means assuming that the image underwent L JPEG compressions with quality factors $QF_1, \dots, QF_L$ in chronological order. A certain sequence of quality factor $\mathbf{QF}$ defines 64 sequences $\mathbf{q}^f = [q_1^f, \dots, q_L^f]$ of L quantization steps, one for each DCT frequency $f = 1, \dots, 64$ and determined according to the quantization tables.
Then, our null hypothesis will be given by the compression chain $[QF_c]$, while alternative hypotheses will be associated to compression chains in the form $[\dots, QF_c]$. The forensic

Figure 4.1: Forensic scenario considered in this Chapter.

task considered is depicted in Fig. 4.1, thus the proposed detector will analyze JPEG images and determine whether they underwent only one compression or other compressions occurred before.

4.2 Benford-Fourier coefficients in JPEG images

As it was introduced in Chapter 3, we can consider BF coefficients as the expected value of the random variable $e^{-j\omega \log_{10} Z}$ and use as unbiased estimator the realization $\hat{a}_\omega$ of the sample mean

$$\hat{A}_\omega = \frac{\sum_{m=1}^{M} e^{-j\omega \log_{10} Z_m}}{M}, \tag{4.1}$$

where Z_m is the random variable representing the DCT coefficient at the chosen DCT frequency in the m-th block (we assume the variables independent and identically distributed between different blocks).

In the case of JPEG images, the values of each random variable Z_m actually fall in a specific set of numbers: the non-negative multiples of the quantization step q_c used for that DCT frequency in the last JPEG compression. Then, the BF coefficient estimates can be expressed as

$$\hat{a}_\omega = \frac{1}{M} \sum_{k=1}^{+\infty} h_c(kq_c) \cdot e^{-j\omega \log_{10} kq_c}, \tag{4.2}$$

where $h_c(n)$ is the number of DCT coefficients with absolute value equal to n. In other words, $\hat{a}_\omega$ is a sum of complex values (in practice, a finite number): in each summand, the phase depends on the quantization step (i.e., it does not vary among images compressed with the same quantization matrix and is known from the object image) and the amplitude depends on the number of DCT coefficients falling in the quantization interval before the last compression, thus being an intrinsic property of each image. By looking at the expression (4.2), we observe that the BF coefficients can be interpreted as a compact representation of a generic DCT histogram, depending on the chosen values of q and ω. A preliminary experimental analysis shows that in the case of discrimination between sin-

gle, double or multiple compression, the magnitude of the BF coefficients is not sufficient for an accurate separation, as it happened in Chapter 3. On the other hand, we can notice that the distribution of the $\hat{a}_\omega$ in the complex contains useful information for our goal. Indeed, the *phase* of the coefficients compressed at the same quality factor presents some regularity. We can observe such phenomenon in Figure 4.2, where we report the complex BF coefficients (frequency $(1, 4)$, $\omega = 8\pi$) of the 1338 images in UCID compressed once or twice at different quality factors.

This suggests to give a statistical characterization of the BF coefficients estimates computed as in (4.2), in order to quantify the likelihood that an object JPEG image was actually compressed just once or it underwent previous compressions in its digital history. To this extent, we are interested in the probability of obtaining a certain realization $\hat{a}_\omega$ of $\hat{A}_\omega$ under the hypothesis that the DCT coefficients at the chosen frequency have been quantized with a certain sequence of quantization steps $\mathbf{q} = [q_1, \ldots, q_L]$ (we drop the superscript f for the sake of simplicity). We will indicate such probability as

$$p(\hat{a}_\omega|\mathbf{q}), \tag{4.3}$$

and in our technique we evaluate it under the null hypothesis that the image is actually single compressed (i.e., $p(\hat{a}_\omega|[q_c])$), and the alternative hypothesis that multiple compressions occurred (i.e., $p(\hat{a}_\omega|[\ldots, q_c])$). In particular, we first predict the value of $\hat{a}_\omega$ according to the specific quantization history and we study the distribution of the error between the prediction and the actual value.

4.2.1 Prediction of BF coefficients

By observing (4.2), we have that, starting from an uncompressed image, we could exactly predict the value of the BF coefficient $\hat{a}_\omega$ for each DCT frequency if the image were successively compressed with a certain quality factor: we have to compute the DCT coefficients from the original image, count the number of elements in each quantization interval according to the chosen quantization table and compute the sum as in (4.2). Similarly, if we have a JPEG image and are able to provide a reliable estimation of the original unquantized DCT coefficients, we can as well obtain a prediction with the same procedure according to the quantization step q_c used in the last compression, and compare it with the real one, computed from the image as in (4.1). Moreover, once the estimated unquantized DCT coefficients (that we will indicate with the vector $\mathbf{y}_{NQ}$) are available, it is possible to predict the BF coefficients also for the case of a sequence of quantization steps $\mathbf{q}$ by applying on $\mathbf{y}_{NQ}$ a repeated quantization[1]. Hence, we define as

$$\tilde{a}_\omega(\mathbf{y}_{NQ}, \mathbf{q}) \tag{4.4}$$

the BF coefficient at ω predicted from $\mathbf{y}_{NQ}$ according to a sequence of quantization steps $\mathbf{q}$.

Clearly, we expect the estimate $\hat{a}_\omega$ to be close (in some sense) to the prediction com-

[1] In this phase, errors due to rounding and truncating operations in the compression process should be considered. This is specified in details in Section 4.3.

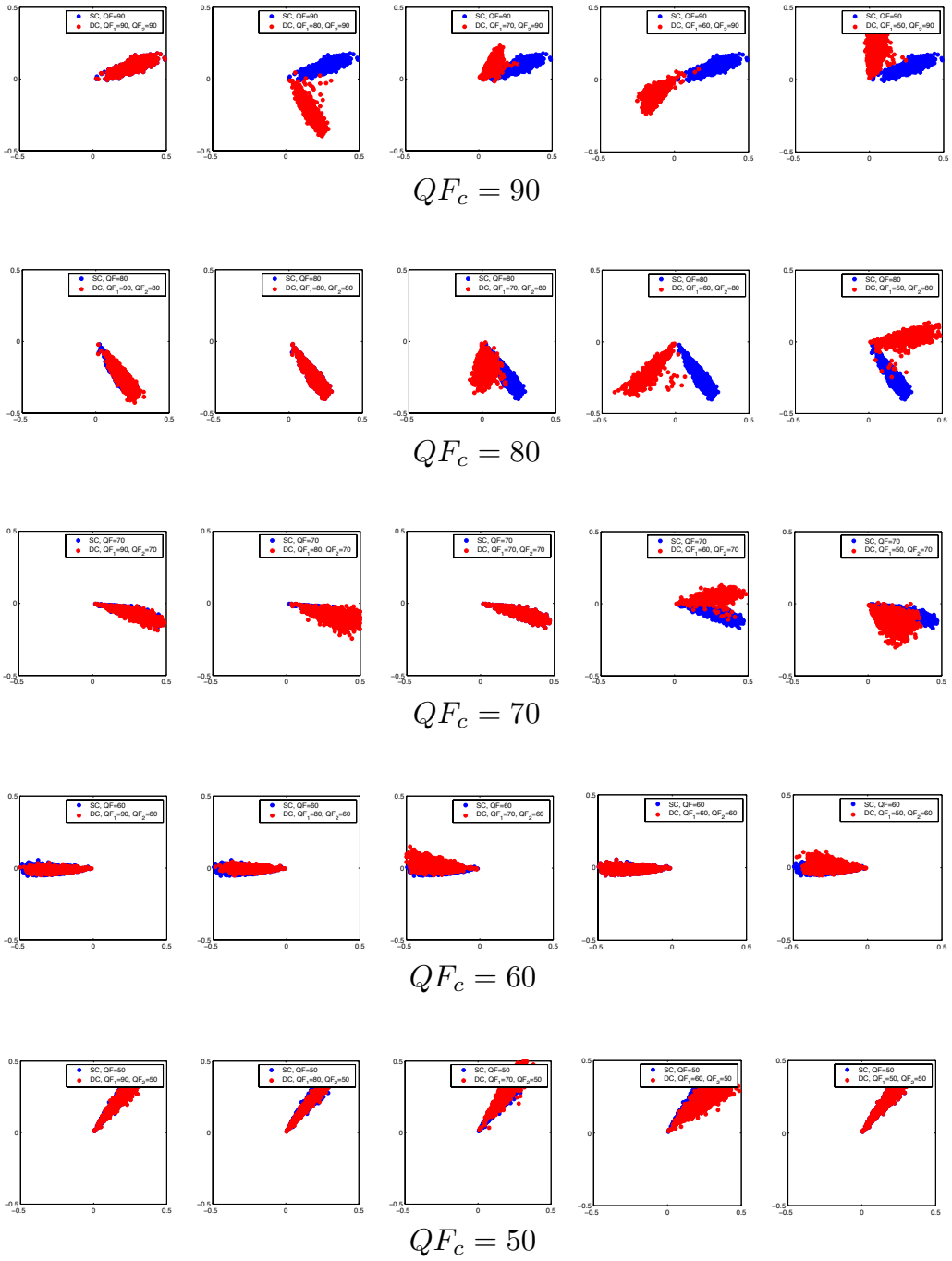


Figure 4.2: BF coefficient estimates for single (blue dots) and double (red dots) compressed images for different QF_c. Primary quality factors $\{50, 60, 70, 80, 90\}$ are used, reported in column-wise order.

puted with the quantization step sequence that was actually used in the image compression history: if the considered frequency was actually quantized only once with step q_c, $\hat{a}_\omega$ will be similar to $\tilde{a}_\omega(\mathbf{y}_{NQ}, [q_c])$; otherwise, if a previous quantization step q_p was applied, $\hat{a}_\omega$ will deviate from $\tilde{a}_\omega(\mathbf{y}_{NQ}, [q_c])$ and lie closer to $\tilde{a}_\omega(\mathbf{y}_{NQ}, [q_p, q_c])$. This behavior will be common to each frequency and can be captured in order to detect previous compressions and estimate the quality factor used.

4.2.2 Prediction error

According to the prediction procedure described in Section 4.2.1, the prediction $\tilde{a}_\omega(\mathbf{y}_{NQ}, \cdot)$ will be in practice affected by an error, depending on the accuracy of $\mathbf{y}_{NQ}$ as estimation of the real unquantized DCT coefficients. We can indicate such error with the complex r.v.

$$E = \hat{A}_\omega - \tilde{a}_\omega(\mathbf{y}_{NQ}, \mathbf{q}), \tag{4.5}$$

so that

$$p(\hat{a}_\omega|\mathbf{q}) = f_E(\hat{a}_\omega - \tilde{a}_\omega(\mathbf{y}_{NQ}, \mathbf{q})), \tag{4.6}$$

where $f_E(\cdot)$ is the pdf of E.

In other words, our goal is to determine $f_E(\cdot)$, that would allow us to compute $p(\hat{a}_\omega|\mathbf{q})$ given the BF coefficient estimate and its prediction, that are available from the image under investigation.

A theoretical derivation of $f_E(\cdot)$ starting from the error on $\mathbf{y}_{NQ}$ is hard to achieve, since the initial deviation of $\mathbf{y}_{NQ}$ with respect to the real unquantized DCT vector is propagated through a complex sum, making it hard to perform an error analysis. However, we can choose a specific technique for the estimation of $\mathbf{y}_{NQ}$ and rely on empirical observations. After preliminary experiments where the calibration technique proposed in [29] has been used, we observed that the prediction error is usually a Gaussian distributed r.v. in the complex plane with different parameters for different values of ω and DCT frequencies, showing a similar behavior even when diverse kinds of images are considered. Starting from that, we adopt a bivariate Gaussian model[2] for the calibration technique, and obtain an estimate of the Gaussian's parameters for a grid of values of ω and all the 64 DCT frequencies, by computing the prediction error in different compression scenarios for a set of images. In particular, for a given ω and a fixed DCT frequency, we have a mean vector $\boldsymbol{\mu}_\omega$ and a covariance matrix $\boldsymbol{\Sigma}_\omega$, so that

$$f_E(e) = \mathcal{N}_2(e; \boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega), \tag{4.7}$$

where $\mathcal{N}_2(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates the bivariate Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Thus,

$$p(\hat{a}_\omega|\mathbf{q}) = \mathcal{N}_2\left(\begin{bmatrix}\mathrm{Re}(\hat{a}_\omega - \tilde{a}_\omega(\mathbf{y}_{NQ}, \mathbf{q}))\\ \mathrm{Im}(\hat{a}_\omega - \tilde{a}_\omega(\mathbf{y}_{NQ}, \mathbf{q}))\end{bmatrix}; \boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega\right), \tag{4.8}$$

[2] As we did in 3.2.2, we treat a complex normal r.v. as a real bivariate r.v., i.e., in terms of variance and covariance of the two single parts instead of complex covariance and pseudo-covariance (as it is usually done when dealing with complex r.v.'s).

An example is reported in Figure 4.3.

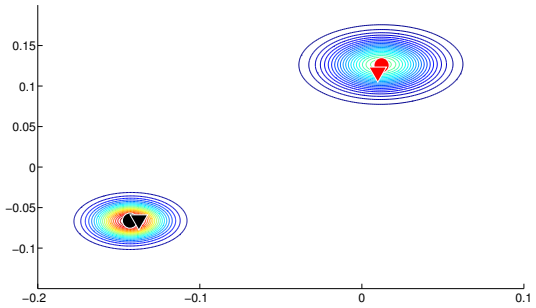


Figure 4.3: A reference image has been compressed with quality factors [80] and [60, 80]; the DCT frequency $(3, 1)$ and $\omega = 27$ have been fixed. In the complex plane, we report the computed values of $\hat{a}_\omega$ and $\tilde{a}_\omega$, represented by filled triangles and circles, respectively. Black markers correspond to the case of compression with [80], red ones to the sequence [60, 80]. Contour plots represent the bivariate pdf's for the two different cases.

4.3 Hypothesis test

Starting from the formulation in 4.1 and statistical analysis developed in 4.2, we can define an hypothesis test where we have a number of alternative hypotheses $\mathbf{H_i}$, $i = 1, \ldots, T$. In particular, the null hypothesis is given by

$$\mathbf{H_0}\text{: the compression chain of the image is } [QF_c]$$

and each $\mathbf{H_i}$ is given by

$$\mathbf{H_i}\text{: the compression chain of the image is } \mathbf{QF_i}$$

where $\mathbf{QF_i}$ is a sequence whose last element is QF_c.

The pool of alternative hypotheses depends on the compression chains we want to test and might include double, triple or multiple compression chains.

In the proposed methodology, we perform a binary hypothesis test for $\mathbf{H_0}$ versus each $\mathbf{H_i}$, $i = 1, \ldots, T$. This is done by computing the likelihood function values $\ell_\mathbf{0}$ and $\ell_\mathbf{i}$ for both the hypotheses, as described in detail in Section 4.3.1. Then, a likelihood ratio is computed for each binary test

$$\lambda_i = \frac{\ell_\mathbf{0}}{\ell_\mathbf{i}}, \tag{4.9}$$

and the final decision is taken as follows

$$\begin{aligned} \mathbf{H_0} \text{ is accepted} \quad & \text{if } \min_{i=1,\ldots,T} \lambda_i \geq 1 \\ \mathbf{H_0} \text{ is rejected} \quad & \text{if } \min_{i=1,\ldots,T} \lambda_i < 1. \end{aligned}$$

The approach is similar to the one proposed in Section 3.3.3 and, in the same way, it works by rejecting the null hypothesis as soon as we find an alternative one which achieves an higher value of the likelihood function. Again, such choice is not derived from a distribution model of λ but it proves to be effective in the forensic scenario considered.

For reasons due numerical stability of the computation of the λ_i, in the following experiments we will equivalently compute the logarithmic likelihood ratio (LLR) [136]

$$\lambda'_i = -2 \cdot \log \lambda_i, \tag{4.10}$$

In this case, the decision rule is then modified as

$$\begin{aligned} \mathbf{H_0} \text{ is accepted} \quad & \text{if } \max_{i=1,\dots,T} \lambda'_i \leq 0 \\ \mathbf{H_0} \text{ is rejected} \quad & \text{if } \max_{i=1,\dots,T} \lambda'_i > 0. \end{aligned}$$

4.3.1 Computation of the likelihood function

As in Section 3.3, we consider a subset F of DCT frequencies to be used and we will add the superscript f to the quantities involved in the following analysis to indicate a specific frequency $f \in F$. In this section, we give a detailed description of a generic binary test $\mathbf{H_0}$ versus $\mathbf{H_i}$ for each $i = 1, \dots, T$.

Starting from the image under investigation, it is first necessary to estimate inherent statistics. First, as mentioned in Section 4.2, we use the calibration technique from [29] (widely used in multimedia forensics [19]) to estimate the unquantized DCT coefficients of the object image, indicated as $\mathbf{y}^f_{NQ}$ for the different DCT frequencies. Second, we consider the noise on the DCT coefficients occurred in consecutive compression processes, due the quantization to 8-bit integers in the pixel domain and the rounding/truncation operations in the computation of the DCT. We adopt the same strategy as in [19] to estimate the parameters of such error, which are used in the following phase. In particular, a simulated noise component is added to the DCT coefficients before every new quantization in the computation of the predictions $\tilde{a}_\omega$ as in (4.4).

Then, we can follow the procedures described in Section 4.2, to compute for a certain DCT frequency $f \in F$:

- the BF coefficient estimate $\hat{a}^f_{\omega^f}$ as in (4.2),
- the predictions $\tilde{a}^f_{\omega^f}(\mathbf{y}^f_{NQ}, \mathbf{q}^f_0)$ and $\tilde{a}^f_{\omega^f}(\mathbf{y}^f_{NQ}, \mathbf{q}^f_i)$ as in (4.4), where $\mathbf{q}^f_0$ is the sequence of quantization steps at f under the null hypothesis $\mathbf{H_0}$ and $\mathbf{q}^f_i$ is the sequence of quantization steps at f under the alternative hypothesis $\mathbf{H_i}$,
- the probabilities of $\hat{a}^f_{\omega^f}$ under the two hypotheses $p(\hat{a}^f_{\omega^f}|\mathbf{q}^f_0)$ and $p(\hat{a}^f_{\omega^f}|\mathbf{q}^f_i)$ by means of (4.8).

The value of ω^f is selected for every frequency so that the Kullback-Leibler divergence between $p(\cdot|\mathbf{q}^f_0)$ and $p(\cdot|\mathbf{q}^f_i)$ (i.e., the respective bivariate Gaussians) is maximized; by doing so, we obtain random variables that are statistically more distant, hence, more easily distinguishable.

Finally, by replicating this procedure for each $f \in F$, we obtain $|F|$ different BF coefficient estimates that we can assume as realizations of independent random variables. Then, we can compute the likelihood functions for each hypothesis as follows:

$$\ell_{\mathbf{0}} = p(\hat{a}^1_{\omega^1}|\mathbf{q}_0^1) \cdot \ldots \cdot p(\hat{a}^{|F|}_{\omega^{|F|}}|\mathbf{q}_0^{|F|}), \tag{4.11}$$

$$\ell_{\mathbf{i}} = p(\hat{a}^1_{\omega^1}|\mathbf{q}_i^1) \cdot \ldots \cdot p(\hat{a}^{|F|}_{\omega^{|F|}}|\mathbf{q}_i^{|F|}), \tag{4.12}$$

4.4 Experimental results

We conducted several experimental tests with two different datasets of uncompressed images, that have been compressed a number of times with an aligned grid according to the considered forensic scenario.

As initial step, we extracted the parameters of the bivariate Gaussian prediction error for each frequency and 100 equally spaced values of ω in $[0, 50]$; in this phase, we used 600 images randomly selected from the UCID database [120] and compressed them with random quality factors, computing the difference between the BF coefficient estimates extracted from the image and the ones predicted as described in Section 4.2.

After this training stage, we then considered the widely studied binary classification problem of single vs double compression, with estimation of the primary quality factor; moreover, we faced the more challenging framework of multi-class classification between single, double and triple compressed JPEG images, where the respective compression chain is estimated.

4.4.1 Single vs Double compression

In this experiment, images compressed once or twice with compression chains composed by quality factors in the set $\{50, 60, 70, 80, 90\}$ are classified by means of the algorithm described in Section 4.3. We used the first 9 non-adjacent DCT frequencies taken in zig-zag order in the 8×8 table.
A pool of 50 possible primary quality factors, $\{50, 51, \ldots, 99\}$ has been considered, thus the alternative hypotheses are given by

$$\begin{aligned} &\mathbf{H_1}\text{: the compression chain of the image is } \mathbf{QF_1} = [50, QF_c] \\ &\qquad\qquad\vdots \\ &\mathbf{H_T}\text{: the compression chain of the image is } \mathbf{QF_T} = [99, QF_c] \end{aligned}$$

The testing set is composed by a subset of UCID dataset (different from the ones used for estimating the noise parameters) and a subset of the Dresden database cropped to their 1000×1000 central part.

In Table 4.1, we report the classification accuracies (meant as accuracy in accepting or rejecting the null hypothesis) for the different last quality factors QF_c and different primary quality factors QF_p obtained by fixing the discriminative threshold for the maximum LLR to 0. From the plots in Figure 4.6, we see that the proposed method generally achieves acceptable results in all the datasets although only UCID images have been used

(a) LIU

QF_c/QF_p	**NC**	**90**	**80**	**70**	**60**	**50**
90	0.76	0.23	1.00	1.00	1.00	1.00
80	0.92	0.85	0.07	0.99	0.99	1.00
70	0.94	0.09	0.93	0.06	0.86	0.99
60	0.93	0.56	0.91	0.66	0.07	0.67
50	0.92	0.11	0.49	0.86	0.27	0.08

(b) UCID

QF_c/QF_p	**NC**	**90**	**80**	**70**	**60**	**50**
90	0.72	0.29	1.00	1.00	1.00	1.00
80	0.85	1.00	0.17	1.00	1.00	1.00
70	0.95	0.57	1.00	0.08	1.00	1.00
60	0.91	0.90	0.99	0.98	0.09	0.99
50	0.97	0.34	0.95	0.99	0.94	0.04

(c) DRESDEN

QF_c/QF_p	**NC**	**90**	**80**	**70**	**60**	**50**
90	0.70	0.26	1.00	1.00	1.00	1.00
80	0.79	1.00	0.19	1.00	1.00	1.00
70	0.90	0.72	0.99	0.07	99.7	1.00
60	0.85	0.91	0.99	0.98	0.13	0.99
50	0.95	0.42	0.94	0.99	0.90	0.03

(d) RAISE2K

QF_c/QF_p	**NC**	**90**	**80**	**70**	**60**	**50**
90	0.90	0.09	0.99	1.00	1.00	1.00
80	0.98	0.99	0.01	0.99	1.00	1.00
70	0.98	0.33	0.98	0.00	0.93	1.00
60	1.00	0.65	0.97	0.90	0.15	0.99
50	0.99	0.15	0.43	0.96	0.69	0.03

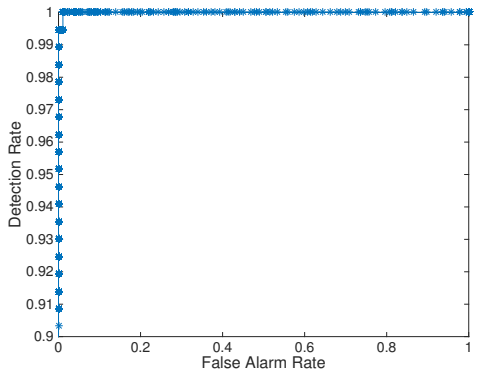
Table 4.1: Accuracies in the different datasets. Secondary quality factors QF_c are placed row-wise and primary quality factors QF_p are arranged column-wise. NC means that only one compression with QF_c has been performed.

in the training stage, proving the robustness of the statistical model for the prediction error. Although some differences among the datasets and the compression chains are observed, the performance is generally good also when the last compression is heavier than the first one, a common limitation of forensic detectors. On the other hand, the discriminative power is very low when the image is recompressed with the same quality factor: this is due to the intrinsic problem that quantization with the same step does not lead to distinguishable DCT coefficient distribution [32], since only rounding/truncation errors during JPEG recompression introduce a difference between once and twice quantized DCT coefficients. This issue will be faced in detail in the next chapter.
Since some higher false alarm rate are observed, we also report in Fig. 4.4 an example of ROC curve for every dataset by varying the threshold on the maximum LLR obtained. We considered the case of single compression with $QF_c = 90$ (which is prone to false alarm) versus the case of double compression with $QF_p = 80$ and $QF_c = 90$. We can see that in any case we could potentially obtain a good performance by tuning the threshold, although that would imply a further training phase.

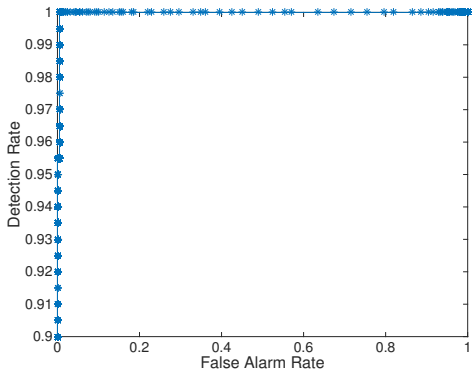
4.4.2 Sensitivity to JPEG implementation

In our tests we employed Matlab built-in JPEG encoder and decoder, although different JPEG implementations exist and are likely used. In order to assess the impact of this specific encoder, we also replicate the computation of the likelihood ratios by using the state-of-the-art libraries `libtiff 3.6.1` and `libjpeg 8d` released by the IJG to read TIFF and write grayscale JPEG images, respectively. In Fig. 4.5, we report the results of this comparison for 200 UCID images. Fig. 4.5a shows the boxplot of the absolute differences between the maximum LLRs obtained with the two encoders in images with

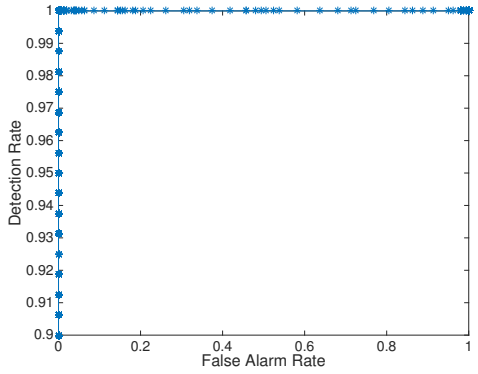
(a) LIU



(b) UCID



(c) DRESDEN



(d) RAISE2K

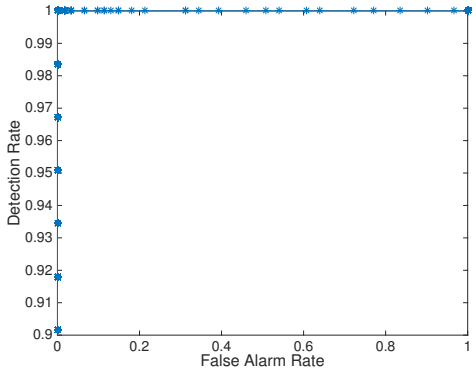
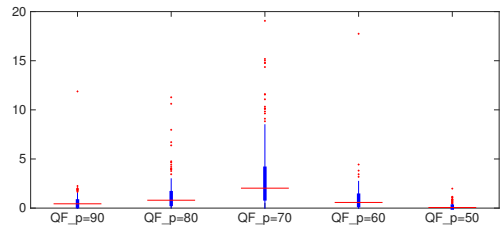


Figure 4.4: ROC curved for the case of $QF_c = 90$ vs $QF_p = 80, QF_c = 90$.

(a) Difference in the maximum LLR for $QF_c = 50$.



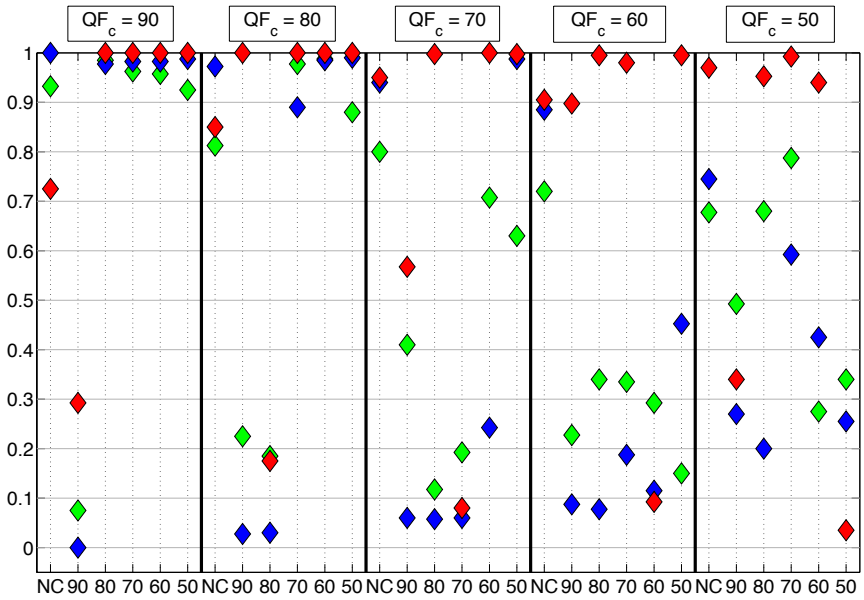
(b) Number of images classified differently.

QF_c/QF_p	**NC**	**90**	**80**	**70**	**60**	**50**
90	0	0	0	0	0	0
80	0	0	3	0	0	0
70	0	0	1	0	0	0
60	0	0	0	1	0	1
50	0	4	1	0	0	0

Figure 4.5: Comparison between `libjpeg` and Matlab encoder.

$QF_c = 50$ and different primary quality factors (reported horizontally). In Fig. 4.5b, we report for each compression chain the number of images that are classified differently when using the two different encoders. We can notice that differences are generally small and do not impact significantly the performance of the method.

(a) UCID



(b) DRESDEN

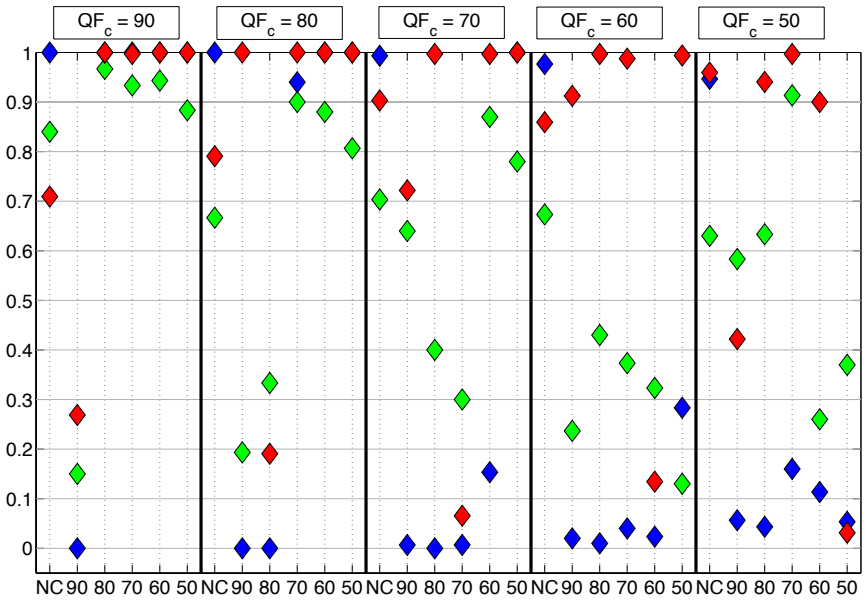


Figure 4.6: Accuracies of the proposed method `BF` (red diamonds) and the techniques proposed in [81] (green diamonds) and [138] (blue diamonds) when varying the compression parameters. The last quality factors used, QF_c, are reported on the top of the plot, while the different primary quality factors are indicated along the horizontal axis (NC means that no previous compression occurred).

4.4.3 Comparison with existing tools

For the sake of completeness, in this section we perform a comparison with existing forensic methods for the distinction of single and double compressed images for the UCID and

DRESDEN datasets. In Figure 4.6, we report the classification accuracies of our method, indicated as `BF`, and the ones proposed in [81] and [138]. The former is based on the Fisher Linear Discriminant Analysis (LDA) using as features the First Significant Digit probabilities of DCT coefficients in the first 20 frequencies and, in order to perform a fair comparison, we train the LDA classifier with the same data used for the extraction of the prediction error parameters. The latter relies on an arithmetic property of DCT coefficients after double quantization, the shape of factor histogram, which is mainly visible when the second compression is lighter than the first one. The technique measures the peaks in such histogram by means of a certain statistic, which is thresholded in order to classify the image as single or double compressed; also in this case, we used the same data as before for determining a proper threshold. For our method, we simply used 1 as discriminant threshold, thus avoiding a further training stage. We can notice the superior performance of the `BF` approach, particularly evident in the case of heavier post-compression.

As a by product of the hypothesis testing scheme, in case of rejection of the null hypothesis we can also obtain an estimate of the compression chain as

$$\hat{\mathbf{Q}}\mathbf{F} = \mathbf{Q}\mathbf{F}_{\hat{i}}, \quad \hat{i} = \arg\max_{i=1,\dots,T} \lambda'_i. \tag{4.13}$$

With this regard, we evaluate also the accuracy in this task and we compare the results obtained with our method and the one proposed in [19], which provide an estimate of the quantization table used in the first compression. In order to obtain the related quality factor, we consider the quantization steps estimated by the method for the first 20 DCT frequencies and find the closest match among a set of known quantization tables related to the quality factors $\{50, 51, \dots, 99\}$. In Tables 4.2 and 4.3, we report the percentage of estimated quality factors with an error lower than 2 with respect to the correct one QF_p (very close quality factors usually have almost identical quantization tables), for the images that are correctly classified as double compressed. Except for the case where $QF_c = 90$, the method `BF` is generally more accurate than [19], especially when the second quality factor is lower.

4.4.4 Single vs Double vs Triple compression

In this experimental setting, we now consider compression chains containing 1, 2 or 3 elements. In this case, in addition to accepting or rejecting the null hypothesis, we also address the task of determining the number of *distinct* compressions occurred, i.e., compressions with not identical quality factors. By doing so, in light of the results obtained in Section 4.4.1, we can assess the power of our method in detecting operation chains that are actually distinguishable.

Precisely, we consider *classes* of compression chains that include chains with the same number of distinct quality factors (up to three) and allowing up to two consecutive compressions with the same quality factor. We denote such classes with bold square brackets. For instance, images compressed as $[70, 80]$, $[70, 70, 80]$, $[70, 80, 80]$ and $[70, 70, 80, 80]$ belong to a single class, indicated as $[\mathbf{70}, \mathbf{80}]$. In our experimental setup, we created the classes by considering all possible pairs and triplets obtained by combining the quality

QF_c/QF_p	Method	90	80	70	60	50
90	BF	0.20	0.99	0.94	0.96	0.84
	[19]	0.00	1.00	1.00	0.99	1.00
80	BF	0.38	0.14	1.00	0.96	0.98
	[19]	0.67	0.00	0.66	0.46	0.91
70	BF	0.85	0.96	0.15	0.98	0.95
	[19]	0.03	0.72	0.00	0.61	1.00
60	BF	0.62	0.06	0.94	0.27	0.92
	[19]	0.00	0.17	0.72	0.00	0.06
50	BF	0.86	0.97	0.55	0.92	ND
	[19]	0.00	0.00	0.06	0.34	0.00

Table 4.2: Primary quality factor estimation for UCID dataset.

QF_c/QF_p	Method	90	80	70	60	50
90	BF	0.31	0.22	0.03	0.01	0.99
	[19]	0.17	0.83	0.90	0.74	0.87
80	BF	0.50	0.25	0.49	0.46	1.00
	[19]	0.24	0.11	0.15	0.22	0.74
70	BF	0.89	0.99	0.29	1.00	1.00
	[19]	0.01	0.25	0.09	0.28	0.95
60	BF	0.58	0.03	0.98	0.30	0.96
	[19]	0.01	0.09	0.36	0.02	0.06
50	BF	0.87	0.99	0.56	0.95	ND
	[19]	0.02	0.00	0.00	0.26	0

Table 4.3: Primary quality factor estimation for DRESDEN dataset.

factors $\{50, 55, \ldots, 90, 95\}$ and fixing as last element the current quality factor QF_c. Then, each alternative hypothesis is related to one class:

$$\begin{gathered}
\mathbf{H_1}\text{: the compression chain class of the image is } [\mathbf{QF_1}] \doteq [\mathbf{50}, \mathbf{QF_c}] \\
\mathbf{H_2}\text{: the compression chain class of the image is } [\mathbf{QF_2}] \doteq [\mathbf{50}, \mathbf{55}, \mathbf{QF_c}] \\
\vdots \\
\mathbf{H_{T-1}}\text{: the compression chain of the image is } [\mathbf{QF_{T-1}}] \doteq [\mathbf{95}, \mathbf{90}, \mathbf{QF_c}] \\
\mathbf{H_T}\text{: the compression chain of the image is } [\mathbf{QF_T}] \doteq [\mathbf{95}, \mathbf{QF_c}]
\end{gathered}$$

Note that the one likelihood value is computed for each alternative hypothesis, according to the chains reported above.

We created 600 testing images from the UCID dataset belonging to 15 different compression chain classes, created by combining quality factors 60, 70 and 80. First, we report the accuracies in rejecting/accepting the null hypothesis for the different classes in Table 4.4.

Then, as in Section 4.4.1 we can considered an estimation of the compression chain class as

$$[\hat{\mathbf{QF}}] = [\mathbf{QF}]_{\hat{i}}, \quad \hat{i} = \arg\max_{i=1,\ldots,T} \lambda'_i. \tag{4.14}$$

This also allows us to estimate the number $\hat{N}$ of distinct compressions occurred as the length of the chains representing the class $[\hat{\mathbf{QF}}]$. For instance, if $[\hat{\mathbf{QF}}] = [\mathbf{95}, \mathbf{70}, \mathbf{QF_c}]$, then $\hat{N} = 3$, and so on.

In Table 4.5, we first report for the 15 classes tested the percentage of images for which $\hat{N} = 1, 2, 3$ (green cells indicate correct classification). The estimation of the number of distinct compression is generally quite accurate, especially when the last compression

Compression chain	Accuracy
[**60**]	0.95
[**70**, **60**]	1.00
[**80**, **60**]	0.45
[**80**, **70**, **60**]	1.00
[**70**, **80**, **60**]	0.90
[**70**]	0.95
[**60**, **70**]	1.00
[**80**, **70**]	1.00
[**80**, **60**, **70**]	1.00
[**60**, **80**, **70**]	1.00
[**80**]	0.95
[**60**, **80**]	1.00
[**70**, **80**]	1.00
[**70**, **60**, **80**]	1.00
[**60**, **70**, **80**]	1.00

Table 4.4: Accuracy for the different compression chain classes (reported row-wise) in accepting or rejecting the null hypothesis.

is lighter. We have errors mostly for double compressed images, that are classified by the method as single compressed (where the primary compression was light) or triple compressed. In the latter case, by looking at the classes detected we notice that, in most cases, an additional very light compression is erroneously identified, for example $[95, 60, 80]$ is detected instead as $[60, 80]$.

Then, in Table 4.6 accuracies on the estimation of the primary quality factors (computed as in Section 4.4.1) are reported, showing that the proposed approach generally achieves good performance also in this task.

4.5 Discussion

We have proposed a method for the detection of previous aligned compression operations in JPEG images. Thanks to several experiments, we assessed its capability of identifying the number of distinct compressions occurred and estimating the quality factors used, obtaining encouraging results. In addition, to the best of our knowledge, this is the first technique allowing for the estimation of the parameters of more than one previous compression operation.

However, we can identify a number of directions for which improvements could be proposed in the future. Firstly, a theory-driven statistical characterization of the prediction error would avoid the need of a training stage. Moreover, we plan to extend the approach based on Benford-Fourier coefficients to a more general case where processing different from aligned recompression are considered, as well as adapt the method in order to allow for the localization of multiple JPEG compression traces.

Tested compression chain/$\hat{N}$	$\hat{N}=1$	$\hat{N}=2$	$\hat{N}=3$
[60]	0.950	0.050	0.00
[70, 60]	0	0.850	0.150
[80, 60]	0.550	0.275	0.175
[80, 70, 60]	0	0.475	0.525
[70, 80, 60]	0.050	0.050	0.900
[70]	0.950	0	0.50
[60, 70]	0	0.550	0.450
[80, 70]	0	0.825	0.175
[80, 60, 70]	0	0.025	0.975
[60, 80, 70]	0	0.025	0.975
[80]	0.950	0.025	0.025
[60, 80]	0	0.575	0.425
[70, 80]	0	0.925	0.075
[70, 60, 80]	0	0.025	0.975
[60, 70, 80]	0	0.025	0.975

Table 4.5: Percentage of images for which $\hat{N} = 1, 2, 3$ is estimated through the 15 different compression chain classes.

Compression chain/Estimation accuracy	$\hat{N}=2$	$\hat{N}=3$
[70, 60]	1.00	
[80, 60]	0.27	
[80, 70, 60]		0.48, 0.86
[70, 80, 60]		0.89, 0.89
[60, 70]	1.00	
[80, 70]	1.00	
[80, 60, 70]		0.79, 0.97
[60, 80, 70]		0.92, 0.97
[60, 80]	0.91	
[70, 80]	1.00	
[70, 60, 80]		0.87, 0.97
[60, 70, 80]		0.90, 0.97

Table 4.6: Accuracy in estimating the primary quality factors on images for which $\hat{N}$ is correctly estimated.

Chapter 5

Double compression traces in high quality JPEG format images

The analysis of multiple compression traces in JPEG images has been faced in the previous chapter. Now, we focus on the more challenging forensic scenario where high quality and nearly identical JPEG compression is applied to the images under investigation in its compression history. The approach developed in Chapter 4 is adapted and integrated with an existing forensic technique that copes with the case of identical recompression. Experimental evaluation has been carried out on the benchmarking datasets.

Acknowledgement

I would like to thank Dr. Pascal Schöttle for the friendly collaboration and Prof. Rainer Böhme for the wise co-supervision of this research, which was conducted during my visiting internship within the Privacy and Security Group of the University of Innsbruck.

5.1 Background

By revising the literature in JPEG forensics, it can be noticed that multiple compression detectors are generally evaluated by considering quite strong quantization, with quality factors typically lower than 90, as we also proceed in Chapter 4. However, the study of high and very high quality JPEG compression is relevant from a forensic perspective, as the availability of memory and bandwidth progressively increases. One may also speculate that counterfeiters store intermediate versions at high quality to avoid visible artifacts and detectable traces.

In addition, the techniques based on DCT coefficients (including the BF analysis) typically fail in detecting previous compressions performed with the same quantization matrix. While some progress has been made in this direction [66, 137, 79, 27], a combination of these techniques with the ones based on DCT coefficients remains unexplored.

For this reasons, in this chapter we focus on the problem of discriminating single and double compressed grayscale images where both the primary and secondary quality factors are larger or equal to 90 and their difference is lower than 10.

Figure 5.1: Forensic scenario considered in this Chapter.

We will see that in such scenario, denoted in the following as *HQ-DC* (high quality double compression) and depicted in Fig. 5.1, difficulties related to the low statistical distance between the different compression chains intensify. In particular, the case of double compression with the same quality factor compromises the performance of the method and force us to rely on another forensic technique (the one based on block convergence in the spatial domain [79, 27]) to cope with that case. The two approaches are then combined by means of decision tree induction theory, complementing the discriminatory capabilities of both the methods.

5.2 HQ-DC scenario

In our forensic scenario, images are single or double compressed. Thus, we can use the same notation described in Section 4.1 by indicating as QF_c the *current quality factor* known by the JPEG file under investigation. The compression chain of each test image will then be composed by a *previous quality factor*, that we can indicate as QF_p, followed

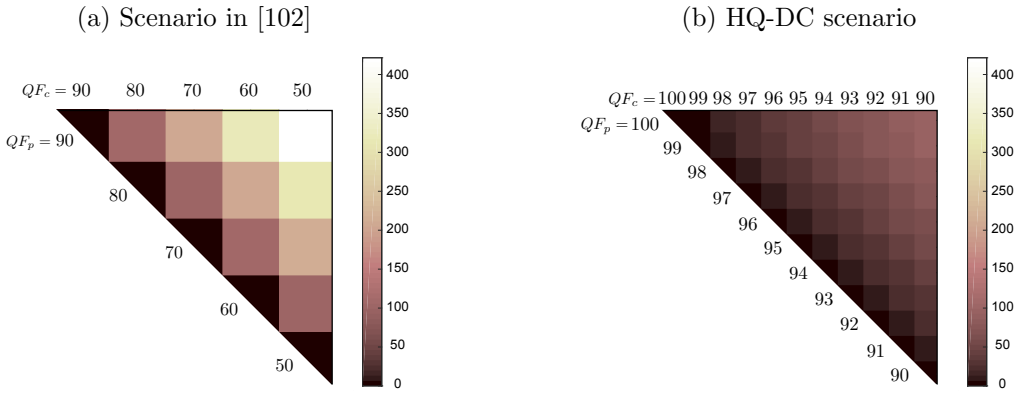


Figure 5.2: Euclidean norm of the quantization table difference in the different double compression chains

by QF_c. For the sake of clarity in the following tests, here we will use the notation $QF_p = NC$ (hence, the compression chain is $[NC, QF_c]$) if the image is single compressed and has no primary quality factor.

In our tests, for each grayscale image $QF_c \in \{90, 91, \ldots, 100\}$ and $QF_p \in \{90, 91, \ldots, 100, NC\}$. Thus, the compression history of the image under investigation can be either $[NC, QF_c]$ (only the last JPEG compression occurred) or $[QF, QF_c]$ (a previous compression occurred), where QF is searched within the set $\mathcal{QF}_{range} = \{90, 91, \ldots, 100\}$.

This kind of setting implies a limited difference between the quantization tables used in the primary and secondary JPEG compression, as well as generally small primary quantization steps which makes the detection of a previous quantization more difficult. Figure 5.2 shows the Euclidean norm of the differences between the quantization tables (luminance channel) of the quality factors considered for the experimental setting in Chapter 4, compared with the ones of the HQ-DC scenario[1].

In our tests, we consider images in uncompressed format and for each of them we create single and double compressed versions by combining all the quality factors in $\mathcal{QF}_{range}$. Thus, each image is processed according to 132 different compression chains, 11 with single compression and 121 with double compression.

For the sake of clarity in describing the tests, we can define the label **SC** for single compressed images and the label **DC** for double compressed images. Indeed, the method in Chapter 4 can be seen as a classifier discriminating **SC** and **DC**. In addition, we should consider that double compressed images can be further divided in images that underwent two compression with the same quality factor and images whose compression chain is composed of two different quality factors. Thus, we can introduce two other mutually exclusive labels indicating these cases, namely **IDC** (identical double compression) and **DDC** (different double compression).

Finally, we can consider the BF analysis as a classifier which in principle is able to distinguish between **SC** (assigned to an image if the maximum value of LLR is below 0), **DDC** (if the estimated compression chain $[QF, QF_c]$ is such that $QF \neq QF_c$) or **IDC** (if the estimated compression chain is such that $QF = QF_c$).

Table 5.1 serves an example of the result representation for our experimental setting. Different colors correspond to different labels of the test images, **SC**, **DDC** or **IDC**. In the following, the numbers in each cell will indicate the classification accuracy with respect to the specific test considered.

We limit our analysis to grayscale images and consequently apply the approach to the luminance channel only. In this experimental setup, we employ the state-of-the-art libraries `libtiff 3.6.1` and `libjpeg 8d` to read TIFF and write grayscale JPEG images, respectively.

5.3 Improved approach

We first present a exploratory experiments on the UCID dataset, from which we develop an improved approach. We use the method as in Chapter 4 and replicate the

[1]We refer to the standard quantization tables used by the `libjpeg` library released by the IJG (Independent JPEG Group), as they are often used in common software.

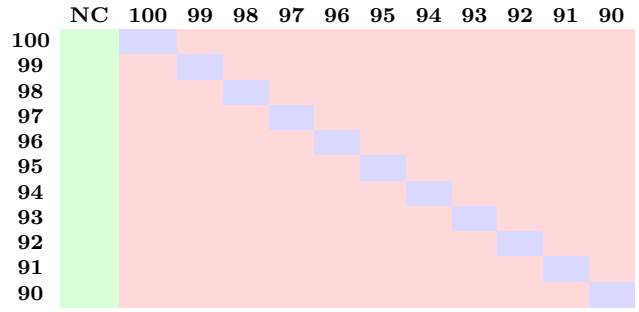


Table 5.1: Example of result representation. Each cell refers to a compression chain specified by the quality factors at the corresponding row and column. Current quality factors QF_c are reported row-wise and primary ones are reported column-wise.

setting adopted in Section 4.4.1, by only changing the set of potential primary quality factor considered as $\mathcal{QF}_{range} = \{90, 91, \ldots, 100\}$. We will indicate such methodology as `BF_baseline`. Let us recall that in Section 4.4.1, a predefined set F of 9 DCT frequencies (specifically $F = \{4, 6, 11, 13, 15, 22, 24, 26, 28\}$ in zigzag order) is used to compute the likelihood ratio (LLR).

We reproduced the experimental setting used in Chapter 4, where a set of 600 UCID images was employed for estimating the prediction error parameters (extended also to other datasets). Such images are then excluded from the Benford–Fourier analysis, while the remaining ones are used for testing.

However, we can observe from Table 5.2 that the performance of `BF_baseline` degrades when moving from the scenario in Fig. 5.2a to the HQ-DC one.

We notice that, although the accuracy for **DDC** is generally high when $QF_p < QF_c$ (the upper triangle of the table), we have a substantial misclassification for **SC**. By exploring the results more closely, we obtain that the Benford–Fourier analysis for images in **SC** generally leads to values of LLR higher than 0 when the alternative hypothesis is given by $[QF_c, QF_c]$. This can be observed in Fig. 5.3, where we report the values of the different LLRs yielded by the quality factors in $\mathcal{QF}_{range}$ for single compressed images.

Moreover, we have high values of the LLR when the primary quality factor tested is high or close to QF_c, even for lower current quality factors. The former phenomenon is due to the small steps used in the primary quantization. The latter one is caused by the fact that

	NC	**100**	**99**	**98**	**97**	**96**	**95**	**94**	**93**	**92**	**91**	**90**
100	0.05	0.98	0.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.01	0.99	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.08	0.04	0.03	0.91	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.21	0.07	0.07	0.65	0.81	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.20	0.03	0.02	0.26	0.90	0.81	0.99	1.00	1.00	1.00	1.00	1.00
95	0.24	0.04	0.08	0.21	0.69	0.71	0.76	0.95	1.00	1.00	1.00	1.00
94	0.27	0.02	0.03	0.41	0.62	0.97	0.78	0.74	0.91	1.00	1.00	1.00
93	0.43	0.06	0.07	0.25	0.71	0.88	0.98	0.83	0.57	0.99	0.99	1.00
92	0.57	0.01	0.04	0.03	0.49	0.17	0.90	0.94	0.77	0.41	0.81	0.96
91	0.54	0.12	0.14	0.75	0.80	0.89	0.98	1.00	0.95	0.76	0.37	0.86
90	0.69	0.05	0.06	0.11	0.26	0.72	0.78	1.00	1.00	0.98	0.71	0.27

Table 5.2: Accuracy of the `BF_baseline` test on UCID dataset.

the quantization tables of QF_c and the one tested might share the very same quantization

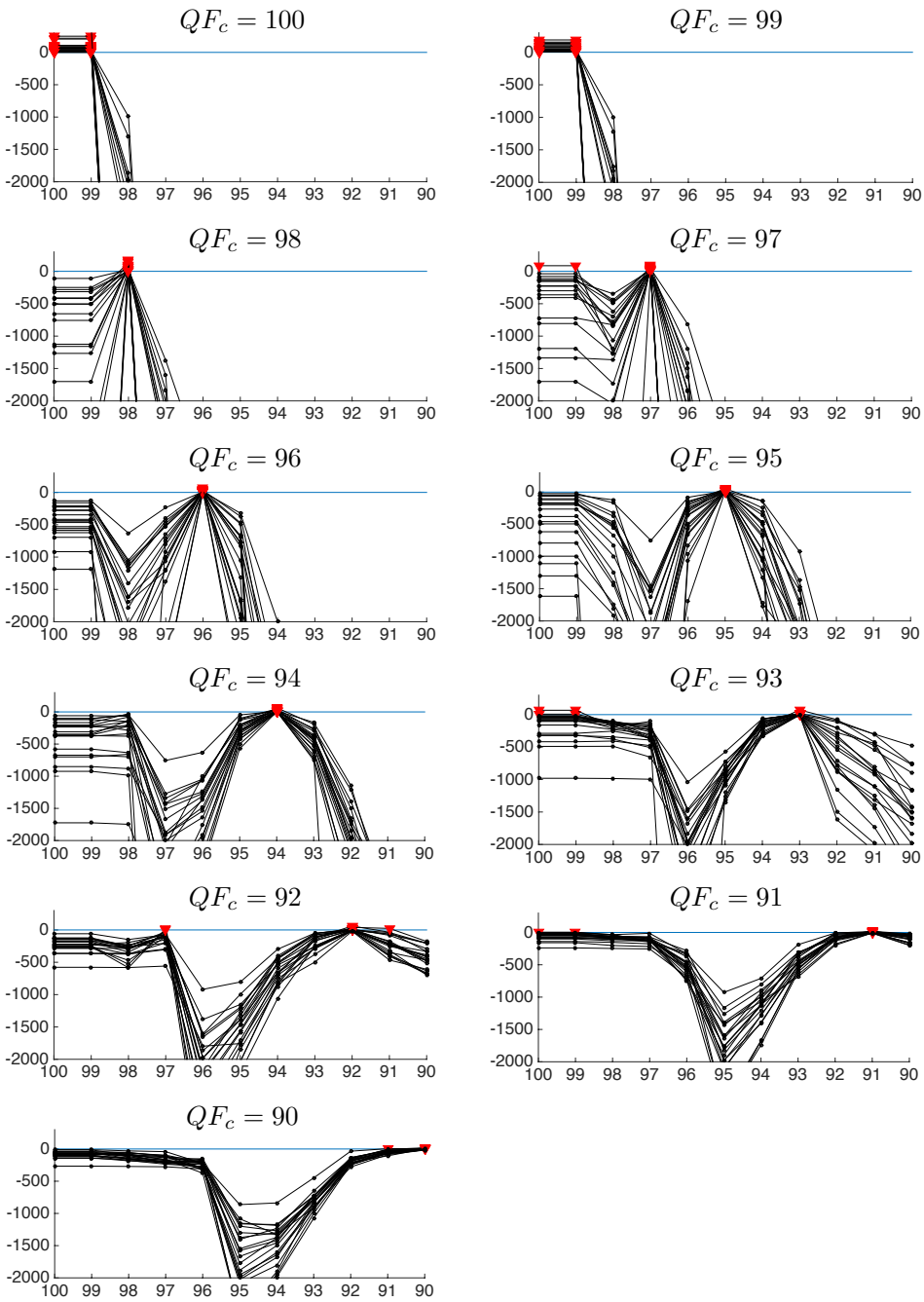


Figure 5.3: Values of the LLRs among different QF_c for 20 randomly selected UCID images. In each plot the horizontal axis contains the quality factors in $\mathcal{QF}_{range}$. The vertical axis represents the value of LLR, that we report in the interval $[-2000, 200]$ in order to compare its behavior across the different QF_c. Each black line corresponds to an image and each value of the LLR that lies above 0 is marked in red.

steps for all or some of the DCT frequencies used in the computation of the LLR, thus decreasing the distinguishability of the two hypotheses. For instance, the quantization table of 99 is equal to 1 up to the 37-th frequency and it fully coincides with the one of 100 at the DCT frequencies used for the computation of the LLR. Thus, when analyzing a JPEG image with $QF_c = 100$, the hypothesis $[99, 100]$ will yield the very same LLR as $[100, 100]$. This likely causes misclassification.

In order to cope with these issues, we design an improved procedure for the selection of the set F of frequencies used. In particular, we propose to adaptively select the set F according to the binary hypothesis test, i.e., choosing the first 9 DCT frequencies in zigzag order among the ones that actually have different primary quantization steps. In the case of $[NC, 100]$ vs $[99, 100]$, the algorithm will choose the frequencies $\{37, 38, 41, 45, 46, 47, 48, 49\}$, where quantization steps for 99 are equal to 2.

This implicitly forces to exclude the hypothesis $[QF_c, QF_c]$ from the pool of alternative ones (as no suitable DCT frequencies would be identified) and to set $\mathcal{QF}_{range} = \{90, 91, \ldots, 100\} \setminus QF_c$. By this, we reduce the misclassification for single compressed images while being aware that the possibility of identical double compression needs to be assessed. In other words, we can consider the resulting new BF test, that we will denote as `BF_adaptive`, as a classifier that discriminates between two classes: if LLR is below 0 for every hypothesis, then images are classified as single compressed or identically recompressed images (labeled as **SC** $\vee$ **IDC**); if at least one hypothesis has a LLR higher than 0, then images are classified as double compressed with different quality factors and labeled as **DDC**.

We report in Fig. 5.4 an example of the different values of LLR obtained with the two different tests, where we can notice the benefit of the frequency selection.

The accuracy results of this approach across the four benchmarking datasets are reported in Table 5.3. For the UCID dataset, the accuracy for **DDC** is unaltered with respect to the baseline approach and misclassification for **SC** is now reduced, although it is no longer distinguished from **IDC** (for this reason the accuracy on **IDC** is also very high). On the other hand, the lower triangle of the table (especially when $QF_p > 95$) remains an issue. It is worth noticing that DRESDEN and RAISE2K show a similar behavior, although accuracy on **SC** and **IDC** is higher. On the other hand, the performance on the LIU dataset presents a significant misclassification for **SC**, that might be due to the reduced size of the images. This suggests that in this case the threshold on the maximum LLR should be tuned ad hoc.

Thus, we can conclude that the Benford–Fourier analysis with adaptive selection of the DCT frequencies is suitable to detect non-identical double compression and is particularly accurate when $QF_p < QF_c$ or $QF_p \leq 95$. This suggests that other techniques can be used to extend the analysis to the detection of identical recompression. To this aim, in the next section we explore the possibility of combining the `BF_adaptive` test with the block convergence approach proposed in [79, 27].

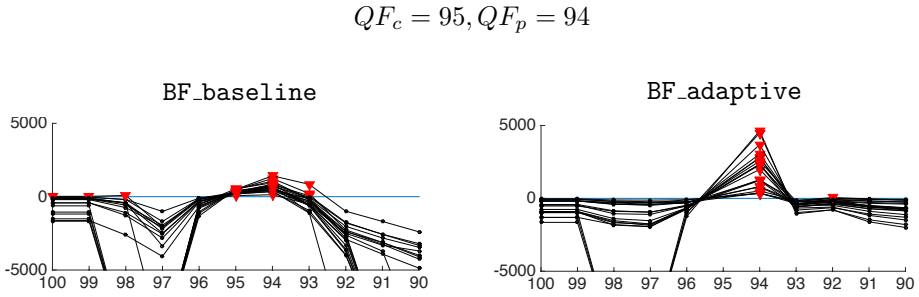


Figure 5.4: Effect of the adaptive DCT selection on the computation of LLR in case of a double compressed image with $QF_p = 94$ (for the `BF_adaptive` test the LLR for $[QF_c, QF_c]$ is not available).

(a) LIU

	NC	100	99	98	97	96	95	94	93	92	91	90
100	0.71	0.29	0.90	0.95	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.57	0.50	0.44	0.95	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.55	0.51	0.54	0.45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.51	0.60	0.63	0.84	0.49	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.57	0.47	0.48	0.66	0.89	0.44	1.00	1.00	1.00	1.00	1.00	1.00
95	0.54	0.51	0.51	0.65	0.90	0.94	0.47	1.00	1.00	1.00	1.00	1.00
94	0.54	0.50	0.55	0.57	0.57	0.97	0.98	0.46	0.99	1.00	1.00	1.00
93	0.50	0.58	0.59	0.66	0.69	0.93	0.98	0.99	0.49	1.00	1.00	1.00
92	0.45	0.64	0.65	0.69	0.69	0.78	0.97	0.98	1.00	0.56	0.98	1.00
91	0.22	0.81	0.85	0.89	0.92	0.86	0.99	1.00	1.00	0.99	0.78	1.00
90	0.34	0.76	0.79	0.79	0.81	0.77	0.96	0.99	0.99	1.00	0.99	0.66

(b) UCID

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.95	0.13	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.94	0.14	0.18	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.91	0.17	0.22	0.79	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.94	0.11	0.12	0.47	0.88	0.95	1.00	1.00	1.00	1.00	1.00	1.00
95	0.92	0.13	0.16	0.36	0.85	0.95	0.93	0.99	1.00	1.00	1.00	1.00
94	0.95	0.15	0.18	0.20	0.20	0.98	0.99	0.94	0.99	1.00	1.00	1.00
93	0.93	0.16	0.19	0.35	0.32	0.96	0.99	0.99	0.93	0.99	1.00	1.00
92	0.94	0.15	0.20	0.24	0.23	0.53	0.96	0.99	0.99	0.94	0.99	1.00
91	0.91	0.16	0.18	0.69	0.79	0.83	0.99	1.00	1.00	0.99	0.90	1.00
90	0.93	0.14	0.19	0.20	0.38	0.44	0.94	1.00	1.00	1.00	0.99	0.93

(c) DRESDEN

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.00	0.00	0.22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.00	0.00	0.02	0.34	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	1.00	0.00	0.00	0.02	0.21	0.84	1.00	1.00	1.00	1.00	1.00	1.00
94	1.00	0.00	0.00	0.00	0.00	0.88	1.00	1.00	1.00	1.00	1.00	1.00
93	1.00	0.00	0.00	0.00	0.00	0.61	0.96	1.00	1.00	1.00	1.00	1.00
92	1.00	0.00	0.00	0.00	0.01	0.02	0.82	0.99	1.00	1.00	1.00	1.00
91	1.00	0.00	0.00	0.10	0.17	0.16	0.89	1.00	1.00	1.00	1.00	1.00
90	1.00	0.00	0.00	0.00	0.00	0.01	0.88	0.98	1.00	1.00	1.00	1.00

(d) RAISE2K

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00
99	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.00	0.07	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.00	0.00	0.59	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.00	0.00	0.28	0.59	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	1.00	0.00	0.00	0.03	0.59	0.83	1.00	1.00	1.00	1.00	1.00	1.00
94	1.00	0.00	0.00	0.00	0.03	0.97	0.97	1.00	1.00	1.00	1.00	1.00
93	1.00	0.00	0.00	0.03	0.00	0.90	1.00	1.00	1.00	1.00	1.00	1.00
92	1.00	0.00	0.00	0.00	0.00	0.31	0.93	1.00	1.00	1.00	1.00	1.00
91	1.00	0.00	0.00	0.41	0.62	0.62	1.00	1.00	1.00	1.00	1.00	1.00
90	1.00	0.00	0.00	0.00	0.14	0.03	0.83	1.00	1.00	1.00	1.00	1.00

Table 5.3: Accuracy of the `BF_adaptive` test across the datasets.

5.4 Combined approach

In this section, we first recall the approach based on block convergence in the spatial domain and some exploratory results obtained with such technique in the HQ-DC scenario (Section 5.4.1). Then, a possible combination approach based on decision tree theory is proposed in Section 5.4.

5.4.1 Block convergence in the HQ-DC scenario

In [79], the authors propose a technique to identify the number of JPEG compressions with quality factor 100 in grayscale images. In this case the quantization table is composed only of the value 1 and we will indicate such setting as JPEG-100. It is observed that for JPEG-100 the 8×8 blocks are transformed in the DCT domain, rounded to the nearest integer and transformed back to the pixel domain, where they are again rounded to the nearest positive integer and truncated to the value range. The authors show that for some of the blocks none of the pixel values change during a JPEG-100 compression. They call these blocks stable and conjecture that after repeated JPEG-100 compression, all blocks of a grayscale image will converge, i.e., become stable. Furthermore, the percentage of blocks that becomes stable after a certain number of recompressions is largely independent of the image content. In particular, given a subject JPEG-100 image, the authors propose to discard the flat blocks (i.e., the ones which contain a single value and are stable from the beginning) and, among the remaining ones, count the ones that become stable after each JPEG-100 recompression. Hence, the ratio of stable blocks (for different numbers of JPEG compressions) is computed by:

$$r = \frac{b_{\text{stable}} - b_{\text{flat}}}{b_{\text{total}} - b_{\text{flat}}}, \tag{5.1}$$

where b_{total} is the total number of blocks in the image, b_{flat} is the number of flat ones and b_{stable} is the number of stable ones. The value of r is then used to identify the number of previous JPEG-100 compressions.

This approach has been extended in [27] to color images, for which all the three color channels need to be analyzed and the block convergence path is studied with respect to a number of additional aspects, such as the kind of color space conversion and the subsampling/upsampling methods. Moreover, in this work the authors propose to fit a theoretical distribution (specifically, the beta distribution) to the ratios of stable blocks observed after different numbers of recompressions. The technique has been tested on images that were recompressed multiple times with the very same quality factor and provided accurate results in case of very high quality images ($QF \in \{100, 99\}$). It also has been noted that the accuracy decreases together with the quality factor.

Now, we want to observe the behavior of the block convergence property in the HQ-DC scenario. To this end, we compute for each UCID image the ratio r as in Equation (5.1), by recompressing the image with the current quality factor QF_c. Then, we used the 600 UCID images discarded in the Benford–Fourier analysis for fitting theoretical models, while the remaining ones are used for evaluating the different tests designed (i.e., the results in the tables refer to the very same images for both methods).

As a first approach, we adopt a maximum likelihood test as proposed in [27], which searches among the pool of potential primary quality factors $\mathcal{QF}_{range} = \{90, 91, \ldots, 100\}$ and is based on a theoretical approximation of the empirical data distribution. In particular, we fit a beta distribution for each of the 132 different compression chains and design a first discrimination test, that we will indicate as `BC_ML`, consisting of the following steps:

- Given an image with a certain QF_c, the value of r is computed.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99
99	0.78	0.53	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.98	0.98
98	0.75	0.27	0.28	0.99	0.76	0.98	0.96	0.97	0.98	0.98	0.99	0.98
97	0.26	0.75	0.76	0.81	0.99	0.95	0.96	0.99	0.98	0.98	0.99	0.99
96	0.38	0.64	0.63	0.69	0.76	0.98	0.91	0.88	0.88	0.97	0.95	0.94
95	0.01	0.99	0.99	0.99	0.99	0.99	0.98	1.00	0.99	0.99	0.99	1.00
94	0.12	0.90	0.89	0.89	0.89	0.88	0.91	0.97	0.95	0.93	0.92	0.89
93	0.10	0.90	0.90	0.90	0.90	0.89	0.90	0.92	0.89	0.96	0.94	0.94
92	0.17	0.82	0.81	0.82	0.82	0.82	0.83	0.83	0.83	0.33	0.89	0.88
91	0.19	0.80	0.80	0.80	0.80	0.80	0.79	0.81	0.80	0.81	0.16	0.88
90	0.07	0.88	0.89	0.89	0.91	0.90	0.90	0.91	0.89	0.89	0.91	0.18

Table 5.4: Accuracy of the `BC_ML` test

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98	0.98	0.98
98	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
97	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	1.00	1.00	1.00	1.00	0.97	1.00	1.00	1.00	1.00	1.00	1.00
95	1.00	1.00	1.00	1.00	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00
94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.96	1.00	1.00	1.00	1.00
93	0.75	0.72	0.72	0.71	0.69	0.77	0.70	0.67	0.96	0.51	0.59	0.61
92	0.03	0.03	0.03	0.02	0.03	0.02	0.03	0.02	0.01	0.99	0.01	0.01
91	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.99	0.00
90	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.99

(a) Accuracy of the `BC_threshold` test with $t = t_1$

	NC	100	99	98	97	96	95	94	93	92	91	90
100	0.99	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
99	0.98	0.96	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
98	0.98	0.98	0.98	1.00	0.72	0.16	0.17	0.16	0.11	0.10	0.05	0.09
97	0.99	0.99	0.99	0.97	1.00	0.78	0.57	0.29	0.35	0.27	0.23	0.22
96	1.00	1.00	1.00	0.99	0.98	1.00	0.83	0.90	0.75	0.26	0.50	0.53
95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	0.95	0.96	0.95	0.77
94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	1.00	1.00	1.00
93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.85	1.00	1.00	1.00
92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.20	1.00	1.00
91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.16	1.00
90	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.11

(b) Accuracy of the `BC_threshold` test with $t = t_2$

- We consider all the beta distribution pdfs $p_{[QF,QF_c]}(\cdot)$ that were previously estimated from every compression chain $[QF, QF_c]$, where QF varies in $\mathcal{QF}_{range} \cup \{NC\}$.
- We evaluate each pdf for r and pick the one that yields the maximum value: if it corresponds to a $QF \in \mathcal{QF}_{range}$ then the image is classified as double compressed, while if it corresponds to the case of NC it is classified as single compressed.

It has to be pointed out that the `BC_ML` is in principle able to distinguish between the three different sets **SC**, **DDC** and **IDC**, as is the `BF_baseline`. However, it also has problems in accurately classifying **SC**, as shown in Table 5.4.

On the other hand the double compressed images are usually correctly identified, in both the **DDC** and **IDC** set (for $QF_c \geq 94$). We can identify the reason of the misclassification for **SC** by looking at the estimated beta pdfs reported in Fig. 5.5. Notice that they are strongly overlapping in the cases of single compression and double compression with $QF_p > QF_c$.

Similarly as it happens for the `BF_Adaptive`, this suggests to reformulate the test with the goal of correctly distinguishing **SC** and **IDC**. In this case, the fitted distribution is clearly separated from the other ones (at least for $QF_c \geq 94$) and represents a relevant open issue for the Benford–Fourier analysis. Moreover, Figure 5.5 also indicates that for $QF_c \leq 93$ almost all of the blocks are already stable. Thus, we would not gain any information by recompressing the image multiple times and get the whole convergence path, as suggested in [27].

Then, we can design a simple threshold-based test (indicated as `BC_threshold`) on r such that an image is classified as single compressed or double compressed with different

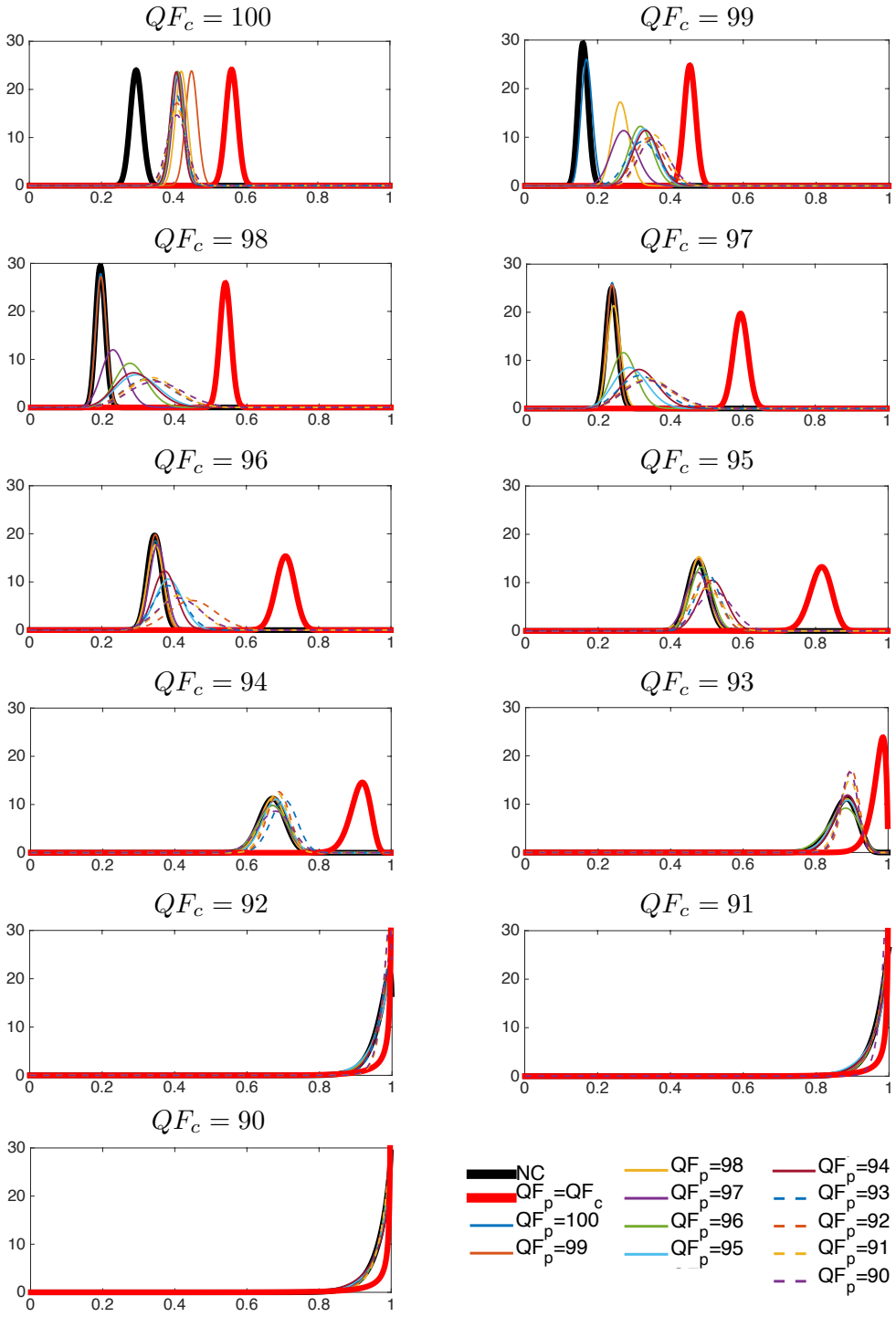


Figure 5.5: Beta distributions of r fitted for different QF_c and QF_p. In each plot the black bold line represents the single compression case with the corresponding QF_c, while the red bold line represents the identical double compression $[QF_c, QF_c]$; other previous quality factors are reported in the legend.

quality factors (**SC** $\vee$ **DDC**) if $r \leq t$, or as identically recompressed (**IDC**) otherwise. The choice of the threshold t can be performed according to different criteria related to the application scenario. As an example, we report in Table 5.5a and 5.5b the results obtained by fixing the threshold for each QF_c in two different ways:

- t_1 is such that $\int_0^{t_1} p_{[QF_c,QF_c]}(r)dr = 0.01$ (we target 99% accuracy on **IDC**),
- t_2 is such that $\int_{t_2}^{1} p_{[NC,QF_c]}(r)dr = 0.01$ (we target 99% accuracy on **SC**).

In practice, we have that both thresholds yield good accuracies when $QF_c \geq 94$ (as it can be expected from Fig. 5.5), while if $QF_c \leq 93$ we have misclassification either for **SC** and **DDC** or **IDC**. Then, we can consider the threshold-based approach on block convergence ratio as accurate for the discrimination of single compressed and identically recompressed images when quality factors are ≥ 94.

In light of the results from Section 5.3, we can notice that the pros and cons of the two techniques are mostly complementary, thus suggesting the development of a combined approach for coping with the HQ-DC scenario. In particular, results from the previous section show that the `BF_adaptive` test distinguishes with good accuracy **SC** $\vee$ **IDC** images from **DDC** images (with misclassification cases for $QF_p > QF_c$); on the other hand, the `BC_threshold` correctly distinguishes **SC** $\vee$ **DDC** images from **IDC** images (with misclassification cases for $QF_c \leq 93$).

The goal is to design a classification test for high quality JPEG images that is able to correctly assign an image to one of the three classes (**SC**, **DDC** or **IDC**) by relying on the knowledge of λ'_M (defined as the maximum LLR value obtained from the Benford–Fourier analysis by excluding QF_c) and r (the ratio of stable blocks after recompressing with QF_c).

This task can be accomplished by means of decision tree induction theory [24, 118], which allows us to determine decision rules on the pair $(\lambda'_M, r) \in \mathbb{R} \times [0, 1]$ obtained from the analyzed image.

5.4.2 Decision tree induction

The decision tree (DT) is one of the most used ways to represent a classification test, which is expressed as a recursive partitioning of the instance space (in our case $\mathbb{R} \times [0, 1]$). The problem of building (or *inducing*) a tree starting from a set of labeled cases (i.e., a set of attribute tuples and corresponding classes) has been extensively studied in the literature [24], providing a number of effective and efficient solutions. A tree is composed by a number of nodes, each of them related to a specific attribute thresholding operation. Nodes that are followed by a subtree are called internal nodes. Otherwise, they are called leaves and they represent the fact that a decision has been reached (i.e, the analyzed attribute tuple has been assigned to a class and no further thresholding is performed). In our case every attribute tuple is a pair $(\boldsymbol{\lambda}'_M, r)$, while the possible classes are **SC**, **DDC** or **IDC**.

In our visual representations of the trees, we will indicate internal nodes as diamonds containing the name of the test used (i.e., `BF` for Benford–Fourier and `BC` for block convergence analysis), while leaves are denoted with squares. We number each internal node

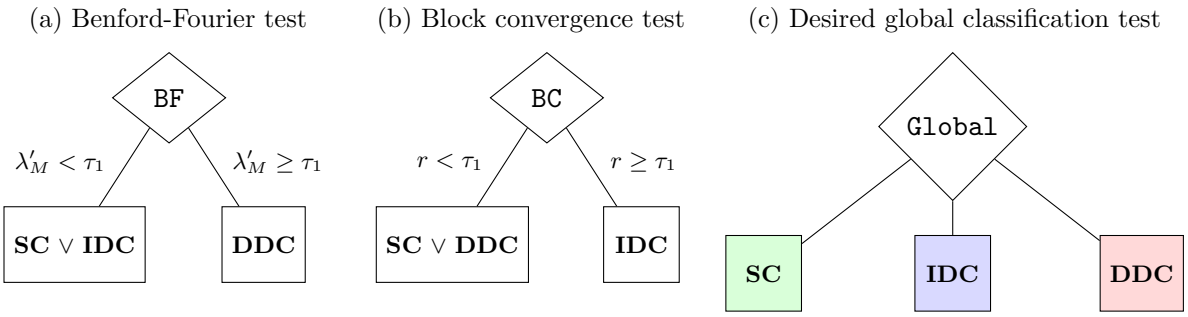


Figure 5.6: Decision trees for different tests.

in top-bottom/left-right order and we denoted as τ_i the threshold used in the i-th internal node. As an example, in Fig. 5.6 we represent the classification tests used in Section 5.3 and 5.4.1, and the one that we want to design.

The induction process generally consists of a *growing* phase, where the tree is developed according to greedy algorithms, and a *pruning* phase, where the tree is further reduced by replacing a subtree with a decision leaf [118]. All these operations pursue the goal of maximizing accuracy on a given training set (a set of samples for which both the attributes and the corresponding classes are known) while minimizing the complexity of the tree, and are performed according to certain criteria and metrics. The result is a list of sequential decision rules that indicates how to optimally threshold the values $\boldsymbol{\lambda}'_M$ and r, and in which order. In the following, DT induction is used to derive accurate classification tests starting from a number of labeled training images.

It is worth pointing out that, differently from other kind of classifiers, decision trees are easy to interpret and represent. Moreover, their complexity can be controlled in several ways, like fixing a maximum number of nodes or a minimum number of leaves. This results in a classification test combining multiple attributes that can be easily conveyed and explained.

5.5 Experimental results

For the sake of brevity, in this phase we consider only UCID (the 738 images used in the previous section) and DRESDEN database, as representatives of smaller and bigger images. For both datasets a number of images has been randomly chosen for the training set of the DT (200 and 100 images, respectively), while the remaining ones are used for testing.

Among the existing toolboxes available for the DT induction, we used the `fitctree` MatLab function contained in the Statistics and Machine Learning Toolbox. We used the default options, with the exception of the prior probability of each class, that we explicitly set uniform. In other words, we consider as equally probable images in **SC**, **DDC** and **IDC**.

Moreover, in each experiment we both determine a *full* DT (i.e., the one that is built in the growing phase) and a *pruned* one, obtained by forcing the algorithm to reduce the tree until it contains less than 8 nodes, in order to have a simplified version.

(a) Training set

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.97	0.10	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.96	0.11	0.15	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.95	0.12	0.17	0.69	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.99	0.04	0.03	0.39	0.79	0.99	0.98	1.00	1.00	1.00	1.00	1.00
95	0.10	0.06	0.06	0.15	0.76	0.91	0.98	0.98	0.99	1.00	1.00	1.00
94	0.01	0.02	0.03	0.04	0.04	0.95	0.99	0.99	0.98	0.99	1.00	0.99
93	0.01	0.01	0.01	0.07	0.12	0.91	0.98	0.98	1.00	0.96	0.98	0.99
92	0.00	0.01	0.01	0.02	0.01	0.35	0.91	0.96	0.97	1.00	0.95	0.99
91	0.00	0.01	0.03	0.43	0.56	0.69	0.97	1.00	1.00	0.96	0.99	0.98
90	0.01	0.00	0.01	0.01	0.06	0.21	0.81	1.00	0.99	0.98	0.93	1.00

(b) Testing set

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.95	0.13	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.95	0.13	0.16	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.94	0.11	0.16	0.83	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.98	0.04	0.07	0.41	0.90	0.98	1.00	1.00	1.00	1.00	1.00	1.00
95	0.01	0.03	0.04	0.17	0.87	0.95	0.99	0.99	1.00	1.00	1.00	1.00
94	0.00	0.00	0.02	0.03	0.04	0.98	0.98	1.00	0.97	1.00	1.00	1.00
93	0.00	0.01	0.01	0.15	0.16	0.96	0.99	0.96	0.99	0.96	0.98	1.00
92	0.00	0.01	0.01	0.02	0.03	0.45	0.95	0.97	0.95	1.00	0.96	1.00
91	0.00	0.00	0.01	0.48	0.68	0.83	1.00	1.00	0.99	0.92	1.00	0.95
90	0.00	0.00	0.00	0.01	0.09	0.28	0.85	1.00	1.00	0.97	0.90	1.00

Table 5.6: Accuracies of overall pruned DT for UCID dataset

5.5.1 Overall decision tree

We first try to build a DT that can be applied to a high quality image regardless of its current quality factor. In this case, the training set is composed of each training image processed according to the 132 different quantization chains, i.e., by using all the quality factors.

The full DTs obtained are quite complex for both datasets, presenting 3137 and 1395 nodes for UCID and DRESDEN datasets, respectively. They are quite accurate on the training set, while the performance strongly degrades when moving to the testing set (see Tables B.1, and B.2 in Appendix B.0.1). This suggests that the algorithm is forced to create a high number of nodes to cope with the specificity of the training set, but it is sensitive to the images it contains. In other words, it suffers from overfitting.

On the other hand, it is worth noticing that the pruned DTs obtained (reported in Fig. 5.7) have the same structure for both datasets (i.e., using BF analysis first to identify **DDC** images and then employ block convergence to distinguish between **SC** and **IDC**), thus differing only in the thresholds used.

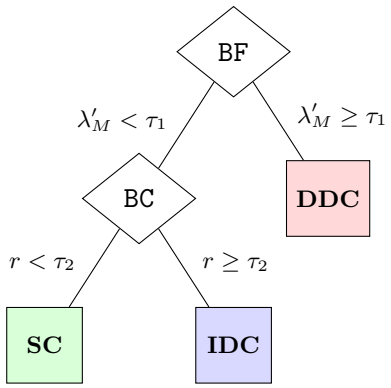


Figure 5.7: Overall best pruned decision tree

Moreover, by observing the accuracies (reported in Tables 5.6 and 5.7) we can notice that the pruned versions achieve good results both in the training and testing set when

(a) Training set

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.99	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.00	0.00	0.20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.00	0.00	0.03	0.36	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	0.00	0.00	0.00	0.03	0.24	0.86	1.00	1.00	1.00	1.00	1.00	1.00
94	0.00	0.00	0.00	0.00	0.00	0.87	1.00	1.00	1.00	1.00	1.00	1.00
93	0.00	0.01	0.01	0.01	0.01	0.66	0.96	1.00	1.00	1.00	1.00	1.00
92	0.00	0.01	0.01	0.01	0.01	0.03	0.83	0.99	1.00	1.00	1.00	1.00
91	0.00	0.01	0.01	0.11	0.14	0.18	0.88	1.00	1.00	1.00	1.00	1.00
90	0.00	0.01	0.01	0.02	0.02	0.02	0.89	0.99	1.00	1.00	1.00	1.00

(b) Testing set

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.00	0.00	0.23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.97	0.00	0.00	0.02	0.34	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	0.00	0.00	0.00	0.03	0.20	0.85	1.00	1.00	1.00	1.00	1.00	1.00
94	0.00	0.00	0.01	0.01	0.01	0.89	1.00	1.00	1.00	1.00	1.00	1.00
93	0.00	0.01	0.02	0.02	0.02	0.62	0.96	1.00	0.99	1.00	1.00	1.00
92	0.00	0.02	0.02	0.02	0.02	0.03	0.84	0.99	1.00	0.99	1.00	1.00
91	0.00	0.02	0.02	0.12	0.18	0.16	0.90	0.99	1.00	1.00	1.00	1.00
90	0.00	0.02	0.03	0.04	0.05	0.05	0.91	0.98	1.00	1.00	1.00	1.00

Table 5.7: Accuracies of overall pruned DT for DRESDEN dataset

QF_c is high, whereas the misclassification for **SC** is noticeably higher for lower values of QF_c in both datasets.

In line with what we observed in Section 5.4.1, these results confirm that the thresholds to be used differ when varying QF_c due to the non-homogeneous behavior of the attributes, especially for r.

5.5.2 QF_c-specific decision trees

Given the results obtained when using the same thresholds for every image, a reasonable solution would be to differentiate the classification test according to the current quality factor, i.e., performing the tree induction process separately for different QF_c. Indeed, it is worth observing that the current quality factor is known, thus such approach is feasible in a realistic forensic scenario.

We repeat the DT building for the 11 different values of QF_c, where the training set is now composed only of images compressed once or twice with QF_c as last quality factor. On the one hand, we have that the full trees have very good accuracies on the training sets but the performance degrades when applied to the testing sets, as shown in Tables B.3 and B.4 in Appendix B.0.2.

On the other hand, pruned trees lead to stable results for training and testing set for both datasets (Tables 5.8 and 5.9). For the sake of brevity, we only report the pruned trees obtained for the different QF_c from the UCID dataset (Fig. 5.8), together with the different thresholds determined in each case. It is interesting to observe how the structure of tree varies among the quality factors, allowing either two or three levels of depth and splitting the nodes in different ways. For instance, we can observe that the first attribute chosen by the algorithm is λ'_M for $QF_c \leq 92$, while it switches to r for higher QF_c for which the block convergence is more accurate.

The results in Tables 5.8 and 5.9 indicate that the pruned DTs determined separately for different values of QF_c yield accurate results. Note that in Tables 5.3, 5.5a and 5.5b the accuracies are computed with respect to the classes discriminated in the two single tests (**SC** $\vee$ **IDC** vs **DDC** for `BF` and **SC** $\vee$ **DDC** vs **IDC** for `BC`). Thus, we can conclude that the capability of `BF` and `BC` of correctly identifying **DDC** and **IDC**, respectively, is generally maintained, while the global misclassification on the three classes is highly

(a) Training set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.04	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.04	0.09	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
97	1.00	0.01	0.03	0.68	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00
96	0.99	0.04	0.04	0.40	0.80	0.99	0.98	1.00	1.00	1.00	1.00	1.00
95	0.99	0.05	0.05	0.15	0.76	0.91	0.99	0.97	0.98	0.99	1.00	0.99
94	1.00	0.02	0.02	0.02	0.04	0.95	0.99	0.98	0.91	0.98	0.99	0.99
93	1.00	0.02	0.06	0.14	0.17	0.93	0.98	0.99	0.86	0.97	0.98	1.00
92	0.99	0.03	0.06	0.07	0.06	0.39	0.92	0.97	0.98	0.36	0.97	1.00
91	0.98	0.01	0.03	0.44	0.56	0.69	0.97	1.00	1.00	0.96	0.28	0.98
90	1.00	0.01	0.02	0.03	0.10	0.24	0.85	1.00	0.99	1.00	0.95	0.23

(b) Testing set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
99	0.99	0.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.99	0.02	0.06	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98
97	1.00	0.01	0.01	0.79	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.98	0.06	0.08	0.42	0.91	1.00	0.99	1.00	1.00	1.00	1.00	1.00
95	0.99	0.03	0.04	0.15	0.86	0.95	1.00	0.98	1.00	1.00	1.00	1.00
94	1.00	0.00	0.01	0.02	0.03	0.98	0.98	1.00	0.92	1.00	1.00	1.00
93	0.99	0.03	0.04	0.20	0.20	0.95	0.99	0.98	0.96	0.98	0.99	1.00
92	0.99	0.02	0.04	0.05	0.07	0.49	0.95	0.99	0.97	0.40	0.97	1.00
91	0.99	0.00	0.01	0.48	0.68	0.83	1.00	1.00	0.99	0.92	0.28	0.96
90	0.99	0.01	0.01	0.01	0.14	0.32	0.88	1.00	1.00	0.99	0.94	0.26

Table 5.8: Accuracies of QF_c-specific pruned DT for UCID dataset.

(a) Training set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.03	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.00	0.01	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.96	0.01	0.04	0.79	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98
96	1.00	0.02	0.01	0.07	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	0.98	0.06	0.10	0.30	0.47	0.93	1.00	1.00	1.00	1.00	1.00	1.00
94	1.00	0.01	0.01	0.01	0.01	0.87	1.00	1.00	1.00	1.00	1.00	1.00
93	0.97	0.10	0.11	0.16	0.13	0.78	0.97	1.00	1.00	1.00	1.00	1.00
92	0.99	0.04	0.07	0.07	0.07	0.08	0.89	0.99	1.00	0.87	0.73	0.99
91	1.00	0.01	0.01	0.11	0.14	0.18	0.88	1.00	1.00	1.00	0.78	0.88
90	0.99	0.04	0.08	0.08	0.10	0.11	0.90	0.99	1.00	1.00	1.00	0.76

(b) Testing set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.99	0.02	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.00	0.01	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.94	0.04	0.05	0.77	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
96	0.97	0.04	0.04	0.06	0.85	1.00	0.99	0.99	0.99	1.00	0.99	0.99
95	0.97	0.06	0.07	0.28	0.36	0.95	1.00	1.00	1.00	1.00	1.00	1.00
94	0.99	0.01	0.02	0.02	0.02	0.89	1.00	0.99	1.00	1.00	1.00	1.00
93	0.94	0.11	0.12	0.14	0.13	0.71	0.97	1.00	0.99	1.00	1.00	1.00
92	0.96	0.07	0.07	0.09	0.08	0.09	0.86	0.99	1.00	0.89	0.68	0.98
91	1.00	0.02	0.02	0.12	0.18	0.16	0.90	0.99	1.00	1.00	0.78	0.85
90	0.97	0.07	0.09	0.10	0.11	0.11	0.92	0.98	1.00	0.99	0.99	0.75

Table 5.9: Accuracies of QF_c-specific pruned DT for DRESDEN dataset.

reduced.

In light of these results, we can consider the QF_c-specific decision trees as a possible effective solution for the distinction of images in **SC**, **DDC** and **IDC** in the HQ-DC scenario.

5.6 Discussion

We have addressed the single vs double compression discrimination problem for grayscale JPEG images compressed with high nearly-identical quality factors (≥ 90). After analyzing the performance of the Benford–Fourier analysis in the DCT domain and the block convergence analysis in the pixel domain, we have studied the problem of combining the two techniques to obtain an accurate discrimination between single compressed images (**SC**), double compressed images with a different quality factor (**DDC**) and images recompressed with the same quality factor (**IDC**).

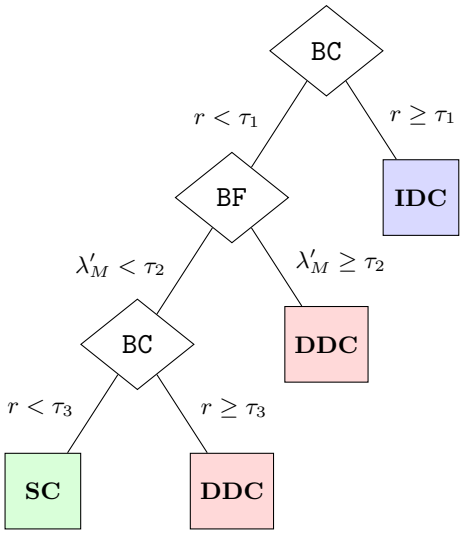
The final set of detectors on both the datasets considered proves to be very accurate for **SC** images (accuracy $\geq 97.5\%$), **DDC** images with $QF_p < QF_c$ (accuracy $\geq 99.0\%$) and **IDC** with $QF_c \geq 93$ (accuracy $\geq 99.5\%$).

The results obtained suggest a number of open issues and directions for future work. The first evident space of improvement is represented by the low detection rate of certain

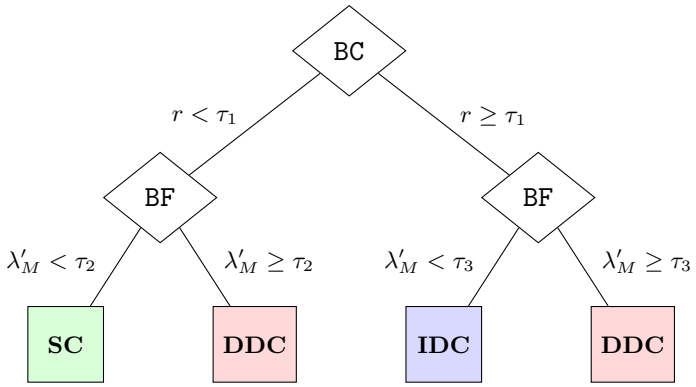
compression chains. For instance, the **DDC** cases where $QF_p > QF_c$ are often misclassified when $QF_p \geq 96$. The same happens for the **IDC** images when $QF_c \leq 92$. Indeed, none of the methods considered is able to correctly identify them and the combined global test does not achieve good performance in those cases, although it leads to improvements with respect to the two separate techniques. As a future perspective, additional methods could be used to cope with these specific issues and incorporated in the final decision tree. For instance, the approaches in [66] or [137] could be employed for identifying **IDC** with $QF_c \leq 92$. Moreover, a limitation of the proposed approach is that it does not explicitly incorporate the knowledge of the size of the image under investigation.

With respect to the process of decision tree induction, currently it is performed by means of standard tools. A potential improvement would be to design induction tools specifically tailored to the forensic scenario considered, by customizing the criteria that rule the construction of the tree. Another, important issue would be to assess the sensitivity to different training conditions, by creating mixed training sets and progressively reducing the number of images. This would also help in building more general decision trees, able to achieve good performance on a generic dataset.

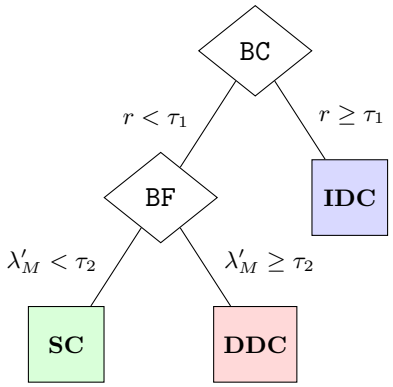
(a) Decision tree for $QF_c \in \{100, 98\}$



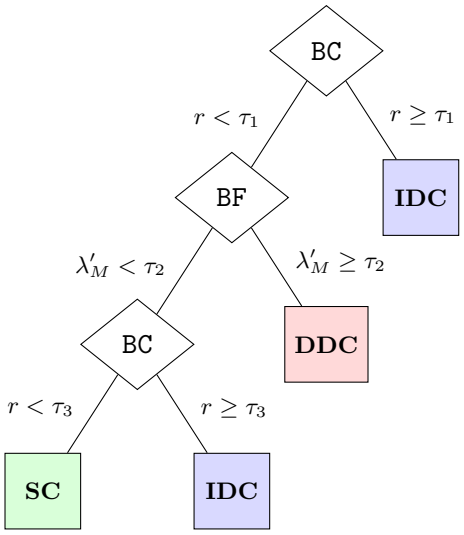
(b) Decision tree for $QF_c \in \{99, 97, 96\}$



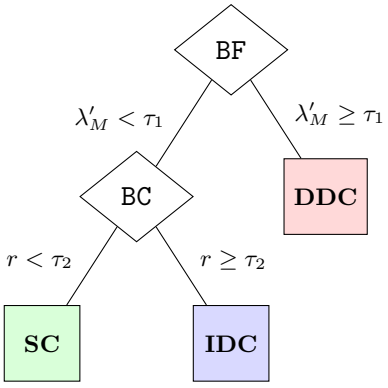
(c) Decision tree for $QF_c \in \{95, 93\}$



(d) Decision tree for $QF_c = 94$



(e) Decision tree for $QF_c \in \{92, 91, 90\}$



(f) Values of τ_1, τ_2, τ_3 for all DTs

QF	τ_1	τ_2	τ_3
100	0.52	324.13	0.34
99	0.41	749.80	5369.97
98	0.47	316.85	0.24
97	0.47	192.85	5088.69
96	0.54	27.56	5033.25
95	0.61	38.13	–
94	0.74	38.46	0.71
93	0.92	19.93	–
92	19.35	0.99	–
91	33.66	0.99	–
90	22.79	0.99	–

Figure 5.8: Pruned decision trees and thresholds for different quality factors for the UCID dataset

Chapter 6

Counterforensics of JPEG compression

Counterforensics is the act of deliberately trying to conceal the traces of data manipulation. In this chapter, two counterforensic techniques are proposed for the reconstruction of statistical properties of natural and JPEG images. They both target the modification of the First Significant Digit (FSD) histogram of the DCT coefficients, in order to conceal traces of single and, in some cases, multiple compression. While the first one indirectly modifies the FSD domain by operating in the modular logarithmic domain, the second one directly targets the reconstruction of a given FSD histogram and can be seen as universal to detectors based on FSD first-order histogram.

Acknowledgement

Part of the work presented in this chapter has been conducted during my visiting internship within the Signal Processing in Communications Group of the University of Vigo. I would then like to thank Prof. Pèrez-Gonzàlez and Prof. Pedro Comesaña-Alfaro for the collaboration and co-supervision.

6.1 Background

By recalling the notation used in Chapter 3 for a certain DCT frequency, we have that

$$X \text{ is the r.v. representing DCT coefficients} \tag{6.1}$$

$$Z \text{ is the r.v. representing the absolute value of nonzero DCT coefficients} \tag{6.2}$$

$$\tilde{Z} = \log_{10} Z \mod 1. \tag{6.3}$$

Clearly, $\tilde{Z}$ is defined in $[0, 1]$, that we will denote as the *modular logarithmic domain*. Alternatively, given a realization z of Z, we will call the correspondent realization $\tilde{z} = \log_{10}(z) \mod 1$ of $\tilde{Z}$ the *logarithmic remainder* of z.
Moreover, let $FSD : \mathbb{R}^+ \longrightarrow \mathcal{D}, \mathcal{D} \doteq \{0, \ldots, 9\}$, be the function mapping any non-

negative real value a to its first significant digit, i.e.,

$$FSD(a) = \begin{cases} \left\lfloor \frac{a}{10^{\lfloor \log_{10} a \rfloor}} \right\rfloor & \text{if } a \neq 0 \\ \\ 0 & \text{otherwise.} \end{cases}$$

Then, we can define the additional r.v. $\hat{Z} \doteq FSD(Z)$, that will play a central role in our methods. While the r.v.'s introduced above are continuous, $\hat{Z}$ is discrete and is defined in the *FSD domain* $\mathcal{D}$.

As remarked in Sections 2.2.1 and 2.2.2, the FSD domain has been exploited for several decision problems in JPEG image forensics. Since the values of DCT coefficients are remapped at any compression step, the distribution of $\hat{Z}$ also changes depending on the quantization factor. Usually, the empirical pmf of $\hat{Z}$ at certain DCT frequencies is analyzed by forensic detectors, as proposed in [59] (uncompressed vs compressed images discrimination), [81] (single vs double compressed images discrimination) and [91] (multiple compression discrimination).

By relying on a known relationship between $\tilde{Z}$ and $\hat{Z}$, our first technique (described in Section 6.2) modifies the realizations of Z (i.e., the DCT coefficients of the image) so that $\tilde{Z}$ follows a certain distribution, in order to reconstruct statistical properties typical of uncompressed images starting from JPEG compression images. It is based on a randomization strategy that guarantees an upperbound on the distortion introduced. The second techniques (described in Section 6.3) tackles the problem of modifying the realizations of Z so that the correspondent realizations of $\hat{Z}$ have a specific pmf (or, equivalently, a specific histogram), trying to minimize the MSE distortion. This is done by formulating the problem as a two-step optimization process and providing a close-to-optimal solution based on heuristic criteria.

6.2 Reconstruction of the modular logarithmic domain statistics

It has been shown in several works [70, 108, 59] that the r.v. $\hat{Z}$ in natural uncompressed images usually follows *Benford's law*, i.e., its pmf is given by

$$P_{\hat{Z}}(d) = \log_{10}\left(1 + \frac{1}{d}\right), \qquad d \in \mathcal{D}. \tag{6.4}$$

With this respect, for a generic r.v. Z, the following property holds [109, 34]:

Proposition 6.2.1 *If $\tilde{Z}$ is uniformly distributed in $[0, 1]$, then $\hat{Z}$ is distributed according to Benford's law.*

In other words, the uniform distribution of $\tilde{Z}$ is a sufficient (not necessary) condition for $\hat{Z}$ to follow Benford's law.

Starting from this result, we design our anti-forensic attack, indicated in the following as `FSD AF`, aimed at modifying the statistics of images that have been compressed once

Figure 6.1: Scenario considered in this Section.

with a known quality factor so that they look never compressed, as depicted in Fig. 6.1. Thus, we suppose the image has been decompressed and is now an uncompressed format file, from which we can compute the 8×8-block DCT. The goal is to modify the current DCT coefficients (which presents traces of compressions) so that their logarithmic remainders are approximately uniformly distributed, thus automatically recovering the Benford distribution in the FSD domain.

As observed also in Section 3.2.2, when computing the DCT from an image stored in uncompressed format that was previously compressed, the absolute values of nonzero DCT coefficients at a certain frequency fall around the multiples of the quantization step q (which is known in this case). We can then consider single quantization intervals $I_k \doteq [kq - q/2, kq + q/2[, k \in \mathbb{Z}^+$, and denote as n_k the number of realizations of Z (absolute values of nonzero realizations of X) falling within each of them. Thus, we define as $\boldsymbol{p}_k, k \in \mathbb{Z}^+$ the vectors containing such n_k coefficients.

Before compression, the elements of $\boldsymbol{p}_k$ were distributed among their quantization interval I_k. The well-known anti-forensic methods in [124] adds a properly distributed noise to any DCT coefficient to recover the original distribution. Differently, from such an approach, we aim at recovering the original uniform distribution of logarithmic remainders, altered during the compression together with DCT coefficients and FSDs.
In particular, for any non-empty vector $\boldsymbol{p}_k$ we consider the values

$$L_1 = \log_{10}(|kq - q/2|), \tag{6.5}$$
$$L_2 = \log_{10}(|kq + q/2|), \tag{6.6}$$

Then, we generate a vector $\boldsymbol{e}_k$ of n_k uniformly distributed values in $[L_1, L_2]$. Vector $\boldsymbol{p}_k$ is finally substituted with its anti-forensically modified version

$$\boldsymbol{a}_k = [\mathbf{10}_{n_k}]^{\boldsymbol{e}_k} \cdot \boldsymbol{s}_k \tag{6.7}$$

where $\mathbf{10}_{n_k}$ is a vector in $\mathbb{R}^{n_k}$ whose elements are equal to 10 and $\boldsymbol{s}_k \in \mathbb{R}^{n_k}$ contains the sign of the corresponding DCT coefficients in $\boldsymbol{p}_k$; the power operation and the vector product are then applied element by element.
It is worth noting that, by doing so we obtain a distribution of the logarithmic remainders

which is *approximately* uniform. Thanks to proposition 6.2.1, this guarantees a Benford's distribution of FSDs. On the other hand, to the best of our knowledge, no theoretic results would assure to get a specific distribution of DCT coefficients (r.v. X in 6.2.1) starting from the distribution of logarithmic remainders, while the vice versa has been done in [109] for the Generalized Gaussian. However, the typical Laplacian shape of the DCT histogram is recovered as well (as shown in Figure 6.2), also thanks to the fact that every coefficient cannot be moved to a different quantization interval. Such phenomenon suggests that the `FSD AF` attack could be effective also when dealing with forensic methods based on the DCT histogram analysis instead of the FSD distribution, as we will show in the next section.

Moreover, this method overcomes a drawback of [124] highlighted in [80]. Indeed, when all (or a very high percentage) of the DCT coefficients in a given subband are quantized to zero, it is not possible to compute the estimation of the Laplacian's distribution parameter. This step is necessary to generate the dither, thus such subbands remain untouched. Following our approach, it is possible to apply the anti-forensic action also in these cases by uniformly randomizing the logarithmic remainders in $]0, q/2]$.
Finally, this procedure can be directly applied to grayscale images, while the three channels in RGB images are treated like in [124].

6.2.1 Visual distortion measure

In order to validate the effectiveness of the proposed anti-forensic action, we first compare our method with the one proposed in [124] in terms of quality of the resulting image. We considered the first 500 images of UCID dataset and compressed them with different quality factors. Then, we applied the anti-forensic attacks (FSD and [124]) and compared the fidelity of the two resulting images with respect to the compressed one. Specifically, we computed for each image the PSNR between the compressed version and the anti-forensically modified image with `FSD AF`, $PSNR_{FSD}$, and the `AF` in [124], $PSNR_{DCT}$. Generally, these values are quite similar (the absolute difference is less than 2 dB for any image) and in most cases the FSD approach gives better results, showing the feasibility of our method in realistic tampering. In Table 6.1, we report the percentage of images for which the difference $D = PSNR_{FSD} - PSNR_{DCT}$ is positive, together with the mean values of PSNR for both the attacks in dB.

QF	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Mean PSNR value for `FSD AF`	33.8	33.7	34.7	34.8	35.3	36.6	39.1
Mean PSNR value for `AF` in [124]	33.2	33.2	34.1	34.3	34.9	36.2	39.1
Percentage of positive D	99.4	98.2	95.2	87.2	78.8	73.0	60.6

Table 6.1: Quality comparison between the anti-forensic attacks

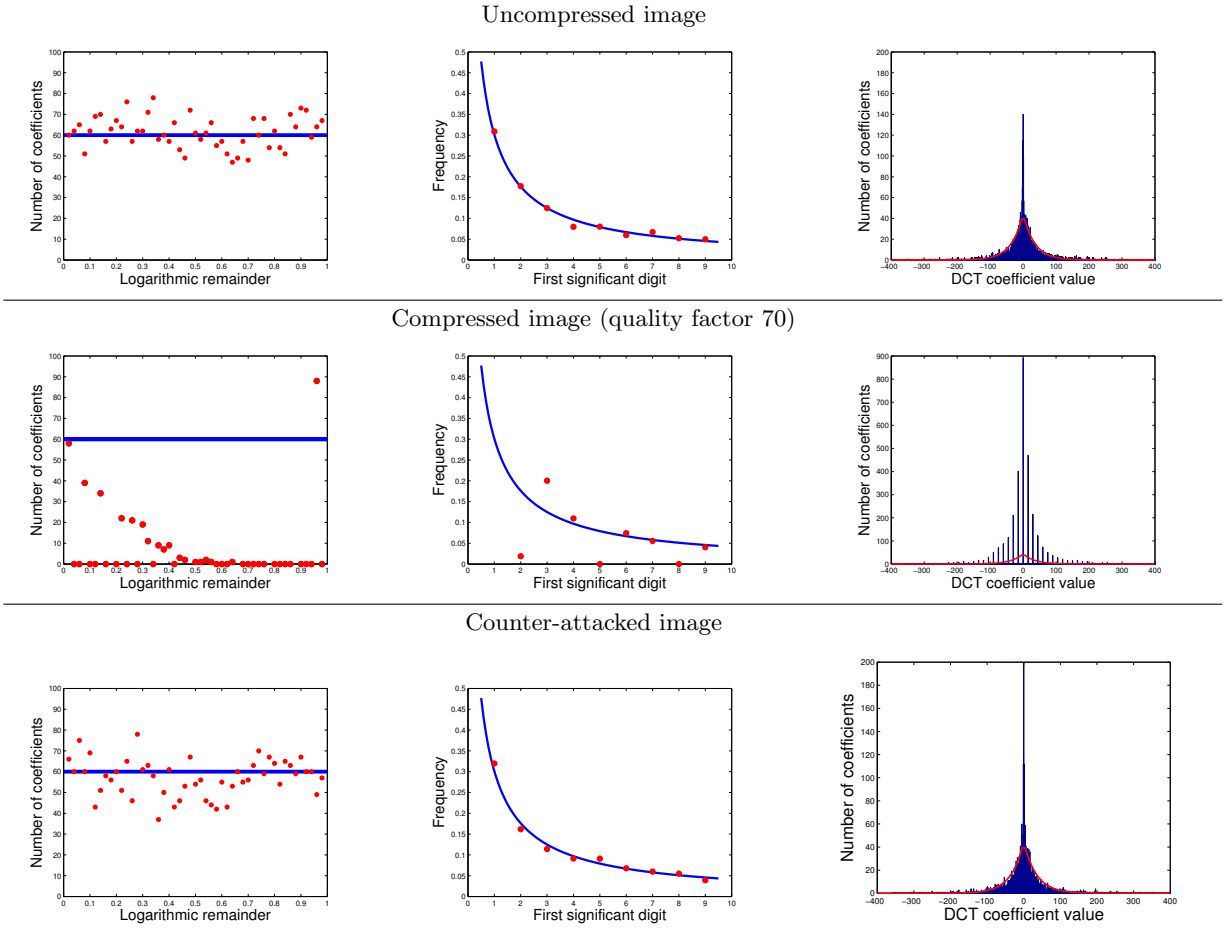


Figure 6.2: Statistics for luminance channel (2,2) DCT coefficients of UCID image 10. Logarithmic remainders and first significant digits are computed for non-zero coefficients. In order to compute the statistics of logarithmic remainders, the interval $[0, 1]$ has been partitioned in 50 equispaced subintervals. In red, number of coefficients whose logarithmic remainder lies within any sub-interval is reported and the blue line represents the ideal uniform distribution, given by the number of subintervals divided by the total number of DCT coefficients. Regarding first significant digits, the curve in blue is standard Benford's law, while frequencies of the nine FSD are reported in red. In the third column, histogram of DCT coefficients are plotted: the curve in red is the Laplacian distribution fitteq starting from the unquantized coefficient like in [124]. Thanks to the uniform randomization of logarithmic remainders, both the Benford distribution of FSDs and the Laplacian distribution of DCT coefficients are accurately restored.

6.2.2 Single versus double compression classification

The first one is the technique proposed by Li et al. in [81], which builds a binary classifier trained to discriminate single and double compressed images by exploiting the FSD frequencies for 20 DCT subbands. The second algorithm, introduced by Bianchi et al. in [20], detects localized forgeries by means of a statistical analysis of DCT coefficients. It is worth noting, that the former one is based on the analysis of FSD distribution of the DCT coefficients, which is exactly the statistics targeted by our approach, while the latter one operates only on the histogram, i.e. the values of the DCT coefficients.

We used the binary SVM classifier designed in [81], trained to distinguish single compressed images from double compressed images, where the quality factor QF_2 of the last compression has been fixed at 80 and the first quality factor QF_1 varies in $[70, 90]$. Primary quality factors QF_1 have been chosen such that $5 \leq |QF_1 - QF_2| \leq 10$, as suggested in [92], in order to avoid a severe degradation of the image but, at the same time, obtain distinguishable data.
We tested the accuracy of the classifier on 138 UCID images (the other 1200 have been used for the training phase), considering the following cases:

- *No AF:* compression with quality $QF_1 \longrightarrow$ compression with quality $QF_2 = 80$
- *FSD AF:* compression with quality $QF_1 \longrightarrow$ proposed FSD anti-forensic attack $\longrightarrow$ compression with quality $QF_2 = 80$
- *AF in [124]:* compression with quality $QF_1 \longrightarrow$ anti-forensic attack in [124] $\longrightarrow$ compression with quality $QF_2 = 80$

Results of the experiments conducted are reported in Table 7.6.

QF1	**70**	**72**	**74**	**86**	**88**	**90**
No AF	75.36%	97.10%	97.10%	96.38%	100.00%	100.00%
FSD AF	10.87%	7.97%	7.97%	1.45%	1.45%	2.90%
AF in [124]	6.52%	10.14%	10.14%	31.88%	39.13%	59.42%

Table 6.2: Accuracy of the SVM classifier in [81]

As a result of the counter-forensic action, the performance of the forensic method strongly decreases, reducing the accuracy of the classifier under 10% in most cases. This proves the effectiveness of the proposed FSD anti-forensics. Moreover, Table 7.6 shows that in all cases (except for $QF_1 = 70$) it outperforms the approach described in [124].

6.2.3 Forgery localization via DCT analysis

After comparing the visual distortion introduced by the two procedures in images belonging to UCID database [120], we considered two forensic methods based on the detection of double quantization artifacts and tested them on the same dataset.

In this second experimental session, we considered the method described in [20]. Here, a probability value of being tampered is computed for each 8×8 block. Since it relies on the assumption that a portion of a single compressed image has been replaced with an uncompressed one and the composite has been finally recompressed, low probability values are assigned to blocks presenting traces of double quantization while single quantized coefficients lead to high tampering probabilities.
In order to assess the effects of our anti-forensic attack and compare it with previous approaches, we applied such algorithm in three different frameworks:

1. compression with quality $QF_1 \longrightarrow$ substitution of the central portion with its original uncompressed version (like in [20]) $\longrightarrow$ compression with quality QF_2

2. compression with quality $QF_1 \longrightarrow$ proposed anti-forensic attack $\longrightarrow$ substitution of the central portion with its original uncompressed version $\longrightarrow$ compression with quality QF_2

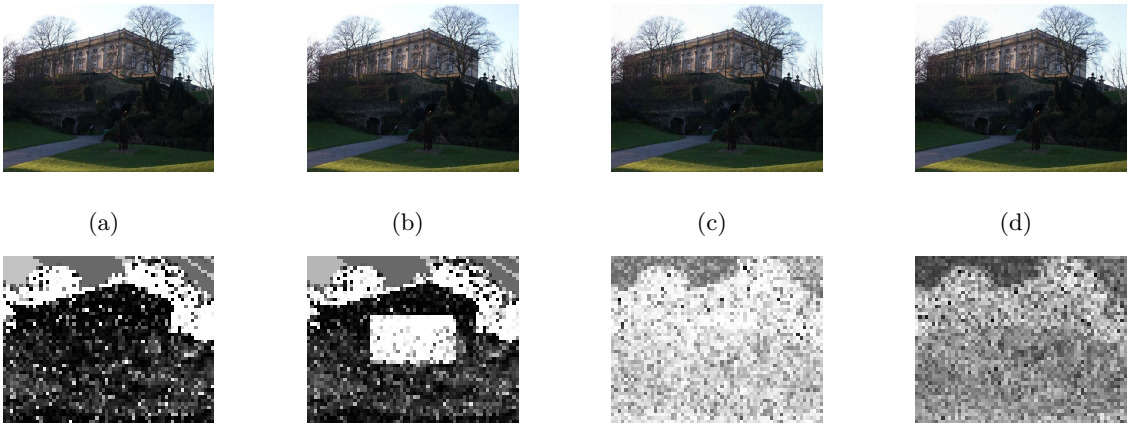
3. compression with quality $QF_1 \longrightarrow$ anti-forensic attack in [124] $\longrightarrow$ substitution of the central portion with its original uncompressed version $\longrightarrow$ compression with quality QF_2

Because of the counter-forensic action, the algorithm should indicate single compression within the whole image, i.e. the probability map should be uniformly high-valued. This would persuade the forensic analyst of the authenticity of the image, since no evidence of tampering would emerge and it would be classified as simply single compressed.
In Figure 6.3, examples of probability maps (for the image 1 in UCID) with $QF_1 = 70$ and $QF_2 = 80$ are shown.
According to [20], in order to evaluate the impact of anti-forensics on the forensic detector performance, in Table 6.3 we report its false alarm rate in the different frameworks, i.e. the percentage of blocks in the double compressed part of the image that receive a probability over 0.5 to be single compressed. Values are averaged over 500 randomly selected images in UCID; quality factors pairs for which best performance is reported in [20] have been chosen.

(QF1,QF2)	**(50,80)**	**(60,80)**	**(70,80)**	**(90,80)**
No AF	8.26%	10.72%	20.31%	23.18%
FSD AF	96.13%	96.62%	95.89%	63.85%
AF in [124]	75.94%	92.48%	93.69%	44.58%

Table 6.3: False alarm rate of the forensic detector in [20]

From Table 6.3 it is evident that the FSD AF attack strongly increases the false alarm rate, i.e. decreases the reliability of the forensic tool. Moreover, its counter-forensic effectiveness is higher with respect to the attack in [124] in any case, especially when the first compression is stronger than the second one.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.3: In (a) the image is simply double compressed, while (b), (c) and (d) were obtained following the procedures described in 1), 2) and 3), with $QF_1 = 70$ and $QF_2 = 80$. Respective probability maps computed with the algorithm in [20] are reported below. No evident visual distortion is introduced by the manipulation, but the probability map in (f) clearly reveals a forgery in the central portion of the image. When anti-forensics is applied (`FSD AF` in (g) and AF by [124] in (h)), uniformly high-valued probability maps are obtained.

6.3 Reconstruction of FSD domain statistics

As previously specified, this technique aims at reproducing a specific pmf $P_{\hat{Z}}$ in the FSD domain. In particular, we focus on the problem of FSD histogram modification and propose a method to replicate a set of target histograms starting from the ones of a given image.

Existing FSD anti-forensic approaches [92] for this specific task present limitations in terms of distortion introduced in the image or lack of flexibility with respect to the forensic scenario. With the same meaning as in [11], our attack can be seen as universal to detectors based on FSD first-order histogram, since no specific method is targeted but we consider a generic binary detector taking as input a set of vectors and analyzing their FSD histograms in order to decide between a null hypothesis H_0 and an alternative hypothesis H_1. Such hypotheses can be adapted to suit different forensic problems coping also with multiple compression, differently from the approach in Section 6.2. In the following, we will always assume hypothesis H_1 is verified for the given image and we want to modify its FSD first-order histogram so that its attacked version leads to the acceptance of H_0. Since a generic detector is addressed, we do not express the acceptance region analytically nor rely on specific models for FSD first-order statistics. However, in our framework we assume to have a reference histogram to be replicated for each frequency, obtained by averaging the histograms of a set of images for which H_0 is verified. Though such assumption might sound restrictive, it proves to be suitable for the forensic scenarios mentioned before. Indeed, it is possible to exploit the fact that the distribution of DCT coefficients at the same frequency is similar among images and so are the FSD first-order

statistics, even after quantization with the same quality factor. Then, we will assume the decision region to be convex and consider as target the histograms obtained from the averaging operation. This leads to a conservative attack, since it is guarantees that the input signal is moved to the acceptance region regardless of the particular detector (thus preserving the universality of the attack), even if a smaller modification might be sufficient.

6.3.1 Problem formulation

In order to use a compact notation throughout the paper, we formally define the First Significant Digits and their histogram.

Given $\mathbf{x} \in \mathbb{R}^N$, with a slight abuse of notation, we will indicate as $FSD(\mathbf{x})$ the vector given by $(FSD(x_1), \dots, FSD(x_N)) \in \mathcal{D}^N$.

Now, we can define the function H mapping any FSD vector $\mathbf{d} \in \mathcal{D}^N$ to a vector $\mathbf{h} \in \{0, \dots, N\}^{10}$, representing its histogram computed by considering bins corresponding to $0, \dots, 9$.

In our case, $\mathbf{x}$ contains all the DCT coefficients, i.e., the realizations of X. Our goal will be to modify $\mathbf{x}$ such that $FSD(\mathbf{x})$ has a certain histogram $\mathbf{h}^*$. In the notation introduced in Section 6.1, it is equivalent to impose a certain pmf on $\hat{Z}$.

As mentioned before, binary forensic detectors proposed in the literature analyze a number of histograms, corresponding to a set of frequencies. However, if a target histogram for each frequency is provided, the process of replicating such histograms can be performed separately on each frequency by means of the same procedure. Therefore, in the following we will consider a single vector $\bar{\mathbf{x}}$, containing the DCT coefficients at a certain frequency, and a given target histogram $\mathbf{h}^*$. We consider a distance g^x defined over $\mathbb{R}^N \times \mathbb{R}^N$ as a measure for comparing $\bar{\mathbf{x}}$ and its modified version.

Then, the problem of modifying $\bar{\mathbf{x}}$ such that its FSD first-order histogram is equal to $\mathbf{h}^*$ minimizing the distortion is equivalent to solving the following optimization problem

$$\mathbf{x}^* = \underset{\{\mathbf{x}|H(FSD(\mathbf{x}))=\mathbf{h}^*\}}{\arg\min} g^x(\bar{\mathbf{x}}, \mathbf{x}). \tag{6.8}$$

In order to express the problem in (6.8) in terms of *optimal transportation theory* [114], as it is done in [33], we define $\bar{\mathbf{d}} := FSD(\bar{\mathbf{x}})$ and a similarity measure for FSD vectors with respect to $\bar{\mathbf{d}}$:

$$g^d(\bar{\mathbf{d}}, \mathbf{d}) := \min_{\{\mathbf{x}|FSD(\mathbf{x})=\mathbf{d}\}} g^x(\bar{\mathbf{x}}, \mathbf{x}).$$

Now, we can state that the solution of (6.8) is equivalent to following sequence of problems:

$$\mathbf{d}^\sharp = \underset{\{\mathbf{d}|H(\mathbf{d})=\mathbf{h}^*\}}{\arg\min} g^d(\bar{\mathbf{d}}, \mathbf{d}), \tag{6.9}$$

$$\mathbf{x}^* = \underset{\{\mathbf{x}|FSD(\mathbf{x})=\mathbf{d}^\sharp\}}{\arg\min} g^x(\bar{\mathbf{x}}, \mathbf{x}). \tag{6.10}$$

It is worth noticing that, unlike in [33], we do not need to optimize over a set of histograms, since we consider a single target one.

However, in order to find the optimal solution, $g^d(\bar{\mathbf{d}}, \cdot)$ must be minimized over all the FSD vectors having histogram $\mathbf{h}^*$. If $\mathbf{h}^* = (h_0, \ldots, h_9)$, the number of vectors to be considered is given by

$$\binom{N}{h_0}\binom{N-h_0}{h_1}\cdots\binom{N-(h_0+h_1+\cdots+h_8)}{h_9}.$$

Such number is generally very high, even for small values of N, thus making the search of the exact $\mathbf{d}^\sharp$ computationally unfeasible.

For this reason, we propose a procedure based on a simple and yet effective strategy, that provides a close-to-optimal solution of (6.9) and (6.10) simultaneously.

6.3.2 Proposed method

As mentioned before, the optimization in (6.9) requires the evaluation of g^d over every element of $\{\mathbf{d}|H(\mathbf{d}) = \mathbf{h}^*\}$. Regarding (6.10), it can be solved more easily if we assume that g^y is a component-wise sum

$$g^x(\bar{\mathbf{x}}, \mathbf{x}) = \sum_{j=1}^{N} g(\bar{x}_j, x_j), \tag{6.11}$$

where g is a symmetric convex function depending on the difference between its input arguments. This is the case of the MSE, the most extensively used distortion measure. Indeed, under these assumptions, minimizing $g^x(\bar{\mathbf{x}}, \cdot)$ is equivalent to minimizing each $g(\bar{x}_j, \cdot)$.

This significantly simplifies solving (6.10). Indeed, if we define $\mathcal{S}$ to be the subset of $\mathbb{R}$ to which we can move the initial values, then $\mathcal{S}$ is the union of the disjoint sets $\mathcal{S}_d = \{s \in \mathcal{S}|FSD(s) = d\}$, $d \in \mathcal{D}$. For any real value y and any digit d, the elements in $\mathcal{S}_d$ that minimize the absolute difference with respect to y can then be identified.

Then, we define

$$f_{\mathcal{S}}(x, d) := \arg\min_{x' \in \mathcal{S}_d} |x - x'|,$$

$$Dist_{\mathcal{S}}(x, d) := |x - f_{\mathcal{S}}(x, d)|.$$

If the optimization problem in (6.3.2) has more than one minimizer, one of them is arbitrarily chosen, thus guaranteeing that $f_{\mathcal{S}}$ is well-defined. For instance, if $\mathcal{S} = \mathbb{Z} \cdot 10^{-1}$ then $f_{\mathcal{S}}(50, 5) = 50$ ($Dist_{\mathcal{S}}(50, 5) = 0$), $f_{\mathcal{S}}(-50, 7) = -70$ ($Dist_{\mathcal{S}}(-50, 7) = 20$), $f_{\mathcal{S}}(50, 3) = 39.9$ ($Dist_{\mathcal{S}}(50, 3) = 10.1$).

Considering this, in the following we propose a sub-optimal approach to solve (6.9), while (6.10) is solved optimally by means of the map $f_{\mathcal{S}}$. The procedure determines a new vector $\mathbf{a}$ starting from a given input vector $\bar{\mathbf{x}}$ and a target histogram $\mathbf{h}^*$. For the sake of simplicity, we assume $\bar{\mathbf{x}}$ is non-negative; otherwise we should just consider the

absolute values of its components and recover the signs of the original vector after the transformation. Our approach relies on the heuristic idea that, in order to obtain a low distortion of $\bar{\mathbf{x}}$, the elements with largest values should be modified as less as possible, since they clearly introduce heavier distortion than the smallest ones. To this end, in our method a suitable new digit is selected for every element of $\bar{\mathbf{x}}$ and each new component is chosen by means of $f_{\mathcal{S}}$, as described below.

Precisely, starting from $\mathbf{h}^*$, we define as $\mathcal{D}_0^t$ the unique set of N elements belonging to $\mathcal{D}$ such that its histogram is $\mathbf{h}^*$.

Then, the input vector $\bar{\mathbf{x}}$ is sorted in descending order by means of a permutation[1]

$$\tilde{\mathbf{x}} = \sigma(\bar{\mathbf{x}}),$$

and, starting from $j = 0$ (greatest value) until $j = N-1$ (smallest value), every component of $\tilde{\mathbf{x}}$ is transformed as follows

$$d_j^+ = \text{argmin}_{d \in \mathcal{D}_j^t} Dist_{\mathcal{S}}(\tilde{x}_j, d),$$
$$a_j^+ = f_{\mathcal{S}}(\tilde{x}_j, d_j^+),$$
$$\mathcal{D}_{j+1}^t = \mathcal{D}_j^t \setminus d_j^+.$$

Finally, the modified vector is given by

$$\mathbf{a} = \sigma^{-1}(\mathbf{a}^+),$$

where σ^{-1} is the inverse permutation of σ, and its FSD histogram is exactly $\mathbf{h}^*$. Therefore, $\mathbf{a}$ and $\mathbf{d} = \sigma^{-1}(\mathbf{d}^+)$ are obtained as approximate solutions instead of the exact ones, $\mathbf{x}^*$ and $\mathbf{d}^\sharp$, respectively.

Because of the sorting operation, for elements with higher values the corresponding new FSD can be chosen among a larger pool of digits; hence, they will be likely kept unaltered or assigned to a digit that leads to a small $Dist_{\mathcal{S}}(\tilde{x}_i, d_t)$. On the other hand, small coefficients might be moved to a new digit that is far from the original one.

Such procedure clearly does not lead to the theoretical optimal solution, since (6.9) is suboptimally solved. However, the fact that the highest values in $\bar{\mathbf{x}}$ (i.e., the ones that would potentially introduce a higher distortion) are mapped to a new FSD such that $Dist_{\mathcal{S}}$ is low, helps to keep a low distortion between $\bar{\mathbf{x}}$ and $\mathbf{a}$. Specifically, such approach will be particularly effective when the values of $\tilde{x}_j$ decay rapidly as j increases, as it happens for DCT coefficients.

6.3.3 Observations and related work

It is worth noticing that, in the framework of JPEG image forensics described before, the vectors $\bar{\mathbf{x}}$ are a transformation of the signal in the pixel domain. However, as pointed out in [33], if the distortion between the provided image and the modified one is measured in

[1]We remark that, since in (6.11) the function g is the same for every j, any permutation that sorts $\bar{\mathbf{x}}$ in descending order can be used (there might be more than one because of repeated values in $\bar{\mathbf{x}}$).

terms of the MSE (or equivalently the PSNR) in the pixel domain, the method proposed in Section 6.3.2 can be applied straightforwardly to $\bar{\mathbf{x}}$. Indeed, the orthonormality of the block-DCT transformation allows us to consider the MSE as a distance directly in the DCT domain, thus satisfying the assumptions on g^x required in Section 6.3.2.

To the best of our knowledge, all of the forensic detectors based on FSD histograms proposed in the literature only consider non-zero-valued DCT coefficients in their analysis, while null coefficients are discarded and the FSD histogram is computed only for bins $1, \ldots, 9$. The formulation in Section 6.3.1 copes with the more general case where also the null coefficients can be moved in order to replicate a target histogram defined over the 10 bins corresponding to $0, \ldots, 9$. However, our procedure can be easily adapted to the 9 bin case by simply defining $\bar{\mathbf{x}}$ as the vector containing the non-zero coefficients at a DCT frequency and considering $\mathcal{D} = \{1, \ldots, 9\}$ when computing the histogram, thus keeping unaltered the null values.

A significant difference of the proposed method with respect to the approach in [92] is that coefficients are moved in sequence depending on their absolute value, and regardless of their initial distribution. Indeed, in such technique, inspired to waterfilling solutions [127], FSD histogram bins with an exceeding or lacking number of elements with respect to the target histogram are first identified and only transfers from the former to the latter ones are allowed. This generally leads to a quite heavy modification in the DCT coefficients, since it reduces the degrees of freedom in the movement of coefficients.

Unlike [100], the procedure proposed here is able to restore any target histogram and it can then be suitable for a larger number of forensic problems. Indeed, the method in [100] imposes a reasonable upperbound to the distance between every coefficient and its attacked version, but it can be applied only for the case where the attacker wants to restore the statistics of uncompressed images, since it does not allow to produce an arbitrary histogram. On the other hand, as long as a reference target histogram is available, the proposed method can potentially be applied in any hypothesis testing problem where H_0 is "image has been compressed n times" and H_1 is "image has been compressed m times". Furthermore, the procedure in [100] only approximately provides a histogram that verifies Benford's law and, especially for high frequencies (or, in general, frequencies were a strong quantization is performed), such approximation can be not accurate. Indeed, it proves to be effective when the lower frequencies are considered, which is true for most forensic methods proposed in the literature (see [81] and [91]), but might not happen for a generic detector.

6.3.4 Experimental results

In order to evaluate the performance of the proposed method, we considered the forensic scenario described in Section 6.3.1, i.e., where a binary forensic detector takes as input a set of FSD histograms corresponding to 8×8 block-DCT coefficients at different frequencies.

The images used in our experiments belong to the UCID database [120].

We considered three different binary hypothesis testing problems, specified in Table 6.4. In each situation, we are interested in modifying the block-DCT coefficients of images

in the decision region corresponding to H_1, in order to be in the decision region of H_0, by introducing a minimal distortion.

	H_0	H_1
A	uncompressed	single compressed
B	uncompressed	double compressed
C	single compressed	double compressed

Table 6.4

Reference sets of FSD histograms have been obtained by averaging the histograms of 600 randomly chosen images in UCID for every frequency, from 1 to 64: specifically, we computed a set $(\mathbf{h}_1^{unq}, \ldots, \mathbf{h}_{64}^{unq})$ from uncompressed images and the families $(\mathbf{h}_1^{QF_t}, \ldots, \mathbf{h}_{64}^{QF_t})$ from single compressed images with quality factors $QF_t = \{50, 60, 70, 80, 90\}$. Then, the averaged histograms have been normalized, so that we have a reference *probability* for each digit, that is transformed into an integer value according to the number of coefficients in each frequency.

A set of images (different of those used for the computation of the target histogram) are applied the processing corresponding to H_1 for the three cases considered in Table 6.4. For each of them, and each frequency, the FSD histogram is modified in order to yield the target FSD histogram. The set $\mathcal{S}$ has been considered, for each frequency, as a lattice with step equal to the maximum over a row of the 8×8-DCT transformation matrix, in order to encompass in the modification the further distortion due to the quantization in the pixel domain. In a first set of experiments we focus on the nonzero coefficients, i.e., $\mathcal{D} = \{1, \ldots, 9\}$.

In Tables 6.5 and 6.6 we report the PSNR corresponding to the average value of the MSE of the modified images with respect to the provided compressed versions (the ones for which H_1 is verified) for each binary decision problem. In order to evaluate the validity of our approach, we implemented the method described in [92] and compared the results obtained when the same target histogram is considered. Indeed, such technique is also designed to replicate a given histogram, thus allowing for a fair comparison with the proposed method. The two methods are denoted in the Tables as `TT` and `WF`, indicating the transportation-theoretic formulation and the waterfilling approach, respectively.

In problem A, images were first single compressed with different quality factors QF_1 and then $(\mathbf{h}_1^{unq}, \ldots, \mathbf{h}_{64}^{unq})$ have been targeted. The same happens in problem B, where images were first compressed with fixed quality factor 75 and then re-compressed with different QF_2. In problem C, images are first compressed with fixed quality factor 75, re-compressed with QF_2 and the histogram sets $(\mathbf{h}_1^{QF_t}, \ldots, \mathbf{h}_{64}^{QF_t})$ were replicated for different QF_t.

As we can see from the tables, the distortion introduced in the image by the proposed method is significantly lower than the one obtained by applying [92]. The difference in terms of PSNR ranges from 3 dB to 12 dB. In a second set of experiments, we included also null coefficients in the modification, i.e., the bin corresponding to 0 is also considered. In this case, the computational complexity significantly increases, since more coefficients need to be moved in every frequency. PSNR results for a subset of images, computed in a similar way to those in Tables 6.5 and

QF_1	50	60	70	80	90
TT	41.03	41.56	42.11	43.38	46.22
WF	34.47	34.22	34.58	34.91	34.77

QF_2	50	60	70	80	90
TT	38.21	38.58	41.33	44.17	42.83
WF	33.14	33.14	34.29	35.86	35.12

Table 6.5: Case A and B, nonzero coefficients, 738 images.

6.6, are reported in Tables 6.7 and 6.8. They are generally different with respect to the previous case, due to the additional constraint on the null values and the availability of more coefficients to be moved, but still we find similar results as before when considering the difference between the two methods.

6.4 Discussion

Two anti-forensic attack to JPEG compression forensic detectors has been proposed.

The first one s applied to single compressed JPEG images and reconstructs the typical distributions of Z, $\hat{Z}$ and $\tilde{Z}$ for uncompressed images. It restores the Gaussian-like statistical distribution of DCT coefficients and the Benford's law distribution of the FSD randomization strategy in a specific domain but it is based on a randomization strategy whose optimality is not discussed.

The second one directly targets the reconstruction of a given First Significant Digit (FSD) distribution and can be seen as universal to detectors based on FSD first-order histogram. Based on heuristic criteria, the technique provides a close-to-optimal solution for the problem of FSD histogram modification with minimal distortion in terms of Mean Square Error (MSE) distortion. Moreover, it can be applied in a more general forensic scenario where statistics after an arbitrary number of compressions is targeted. However, the distribution of DCT coefficients is not controlled.

We can also perform a comparison between the two methods (`FSD AF` and `TT`), although some observations are in order. Indeed, `FSD AF` leads to a FSD histogram that depends

QF_t	QF_2	50	60	70	80	90
50	TT	42.71	41.37	38.49	39.20	38.61
	WF	38.43	37.32	36.33	35.03	36.05
60	TT	42.08	42.95	41.33	39.08	40.70
	WF	33.65	38.20	36.49	36.11	36.55
70	TT	39.23	42.25	42.98	42.16	43.57
	WF	32.34	32.77	37.66	37.36	37.45
80	TT	36.99	38.93	41.93	44.04	43.82
	WF	32.17	32.40	33.31	39.13	34.57
90	TT	36.17	36.73	39.22	44.16	40.85
	WF	32.99	32.46	33.41	35.14	34.35

Table 6.6: Case C, nonzero coefficients, 738 images.

QF_1	50	60	70	80	90
TT	43.67	43.67	43.60	43.54	43.43
WF	37.67	37.79	37.71	37.93	36.98

QF_2	50	60	70	80	90
TT	43.23	43.19	43.58	43.60	43.63
WF	34.79	34.36	36.88	38.45	37.78

Table 6.7: Case A and B, zero coefficients included, 300 images.

QF_t	QF_2	50	60	70	80	90
50	TT	34.36	34.26	34.79	36.05	35.28
	WF	33.66	31.63	31.98	32.30	32.28
60	TT	35.03	34.89	35.47	36.45	35.96
	WF	31.03	33.67	32.17	32.33	32.34
70	TT	35.66	35.76	36.45	37.56	37.04
	WF	30.69	30.81	34.43	32.74	33.05
80	TT	36.06	36.49	37.79	39.41	38.74
	WF	30.82	31.13	31.54	34.81	32.27
90	TT	37.12	37.14	39.36	42.88	40.71
	WF	31.08	30.98	32.09	33.23	32.87

Table 6.8: Case C, zero coefficients included, 100 images.

on the input signal and is obtained by means of a random process and it is not targeted to optimality. However, in order to compare the quality of the resulting image in a realistic forensic scenario, we applied both approaches to the first 20 DCT frequencies only, as state-of-the-art forensic detectors limit their analysis to these frequencies. In particular, we consider single compressed images and target the statistics of uncompressed images. The behavior of the average MSE (whose corresponding PSNR is reported in Table 6.9) varies together with the quality factors. This is due to the fact that, when a heavier quantization is performed, the two methods restore histograms that are not very close for frequencies 15-20, because `FSD AF` exploits a random signal-dependent process while we impose a conservative reconstruction. On the other hand, when quantization is lighter, the histograms almost coincide and our strategy leads to a better quality in the resulting image.

QF_1	50	60	70	80	90
TT	44.58	46.00	47.71	50.46	54.50
FSD AF	45.18	46.16	47.80	49.99	53.65

Table 6.9: Comparison of `FSD AF` and `TT`.

Both the techniques represent a further threat to the reliability of forensic analysis and an incentive for the development of enhanced forensic tools in an adversary-aware perspective.
The analysis of the traces left by such counter-forensic action and their detectability by means of forensic methods would represent an interesting direction [132, 80]. Moreover,

it would be of great interest to extend our approach to distortion measures different from the MSE (i.e., the PSNR), such as the SSIM or the WPSNR.

Chapter 7

1D median filtering: an example of deterministic forensics

Differently from the approaches presented in the previous chapters, the forensic technique presented in this final chapter is based on deterministic properties. It is targeted to the detection of a median filter application in 1D data. The method relies on mathematical properties of the median filter, which lead to the identification of specific relationships among the sample values that cannot be found in filtered sequences. Hence, their presence in the analyzed 1D sequence allows excluding the application of the median filter. Owing to its deterministic nature, the method ensures 0% false negatives and, although false positives (not filtered sequences classified as filtered) are theoretically possible, experimental results show that the false alarm rate is null for sufficiently long sequences. Furthermore, the proposed technique has the capability to locate with good precision a median filtered part of 1D data and provides a good estimate of the window size used.

7.1 Background

As a result of globalization and worldwide connectivity, people from all over the planet are exchanging ever increasing amounts of information of whatsoever type and form, in the most diverse fields of human activity including science, economy, social relationships, news, entertainment. Forensics technologies provide powerful tools to verify the authenticity of data and their possible manipulation, either they refer to multimodal sensed signals, medical records, geophysical observations, marketing statistics or financial reports. In this framework, the detection of any operation that could have been employed to post-process a set of data, either for malicious purposes or simply to improve their content or presentation, turns out to be of interest for a comprehensive forensic data analysis.

Here, we consider the median filter [128], a widely known technique commonly used for data smoothing. Thanks to its ability to effectively discard outliers while preserving relevant information, median filtering has been extensively adopted as a post-processing operator in different fields, including audio processing [113, 72], image processing [4, 94],

geophysics [85], economics [135], biomedical signal processing [99], both for 1D and 2D signals. Several methods have been proposed for the forensic detection of median filtering, with particular attention to images. Most of them are based on a statistical characterization of the filtered signal in different domains, often relying on machine learning tools for the detection.

To the best of our knowledge, no specific methods for the forensic analysis on 1D data are available, as they usually focus on the two-dimensional case, although part of them can be easily conceived and adapted to the 1D domain and will represent a benchmark comparison in our experimental validation phase. In [77] Kirchner and Fridrich proposed a simple yet effective median detector that exploits the artifacts introduced by the filter. The ratio of histogram bins and the subtractive pixel adjacency matrix features in the first-order difference domain are used as traces to detect median filtering in bitmap images. The first-order difference map is employed in [25] by Cao et al. to compute the probability of zero-values in texture areas of the image. A more complex median detector was proposed by Yuan in [139], based on the idea that median filtering, applied to overlapping blocks, affects the pixels ordering in each block, thus introducing a strong dependence between median values originating from overlapping filter window. Kang et al. in [71] analyzed the statistical properties of the median filter residual by using an autoregressive model. In [31] Chen et al. proposed an effective median detector based on two sets of features, the cumulative distribution function of k-th order image difference (global probability) and the local correlations between different adjacent image difference pairs (local correlation). Recently, an effective median forensic algorithm was proposed in [141], where the second-order local ternary patterns are used to capture the changes of local textures due to median filtering. All of the above techniques rely on statistical classification as a final step of the detector, thus producing both false alarms and missed detections.

On the other hand, we propose a deterministic approach that exploits some basic mathematical properties of the median filter, which are a consequence of its very definition and enforce specific relationships among the samples of the original and filtered sequences. Such properties lead to the identification of sets of 1D patterns (called in the following *unfeasible classes*) that cannot be output by a median filter. While the problem of identifying roots of median filters (i.e., patterns that are certainly preserved after the filtering) has been widely addressed in the past [61], to the best of our knowledge this is the first work focusing on the study of patterns that are certainly not introduced by a median filter.

This turns out to be particularly important for forensic purposes, as the subject signal can be scanned sequentially and the presence of such patterns can be checked by means of a simple algorithm: if they are detected, then the signal is classified as negative to median filtering; otherwise, it is classified as positive to median filtering. It is worth pointing out that the detection does not require any thresholding or training operation, as it is a direct result of the scanning procedure. Moreover, given the nature of the algorithm, the rate of false negatives is guaranteed to be null, as no unfeasible classes can be present in filtered sequences.

The effectiveness of the technique is proved by extensive experiments on different kinds of 1D data, including audio tracks, economical series, physiological signals. Besides con-

firming the absence of false negatives, experimental results demonstrate that in practical cases the method easily achieves 0% false positives, which would be possible in principle. Indeed, the occurrence of the unfeasible classes in common data originated from different sources is extremely frequent. Moreover, the detector is able to provide as a side-information the size of the applied median filter and, thanks to the capability of detecting the unfeasible classes throughout the entire sequence, the technique can be used to segment with a high precision the filtered subsequence in the case of local filtering. Although the proposed scheme is deterministic when the median filter is the very last process applied, we also explored the possibility to exploit the distribution of the unfeasible classes in the signal to detect median filtering even when a post-processing operation is applied, thus addressing robustness issues of the deterministic detector.

7.2 Median filter detection and unfeasible sequences

In this work, we design a forensic detector of median filtering for 1D data based on deterministic properties of such processing, which can be applied to 1D signals or, in general, to any set of ordered one-dimensional data samples.

First, we introduce the theoretical background and the main rationale behind our method in Section 7.2.1. Then, in Section 7.2.2 we propose an algorithmic procedure for the analysis of 1D data, that will be exploited in the following sections.

7.2.1 Theoretical background

We will represent the one-dimensional objects analyzed as numerical sequences $\{y_i\}_{i=1}^{\infty} \subset \mathbb{R}$, that we will simply denote as $\{y_i\}$ for the sake of brevity[1] . Then, the action of median filtering can be defined as follows:

Definition *Given a sequence $\{x_i\}$ and a natural odd number N, the output of a median filter with size N applied to $\{x_i\}$ is a sequence $\{y_i\}$ such that*

$$y_i = median(X_i),$$
$$X_i := \{x_{i-\lfloor \frac{N}{2} \rfloor}, \ldots, x_i, \ldots, x_{i+\lfloor \frac{N}{2} \rfloor}\},$$

when $i > \lfloor \frac{N}{2} \rfloor$ and $y_i = 0$ otherwise.

In a forensic framework, we assume to analyze a 1D data and look for traces of previous median filtering. In other words, we deal with an "inverse" problem, where we are given a sequence $\{y_i\}$ and we need to determine whether it is the output of a median filter of a certain size N applied to an original unknown sequence $\{x_i\}$ or not.

In order to provide an answer to this question, we can now exploit the following consequence of the median filtering definition:

[1] In practice the sequences to be analyzed will be finite, thus we will have that $y_i = 0$ when i is higher than a certain value.

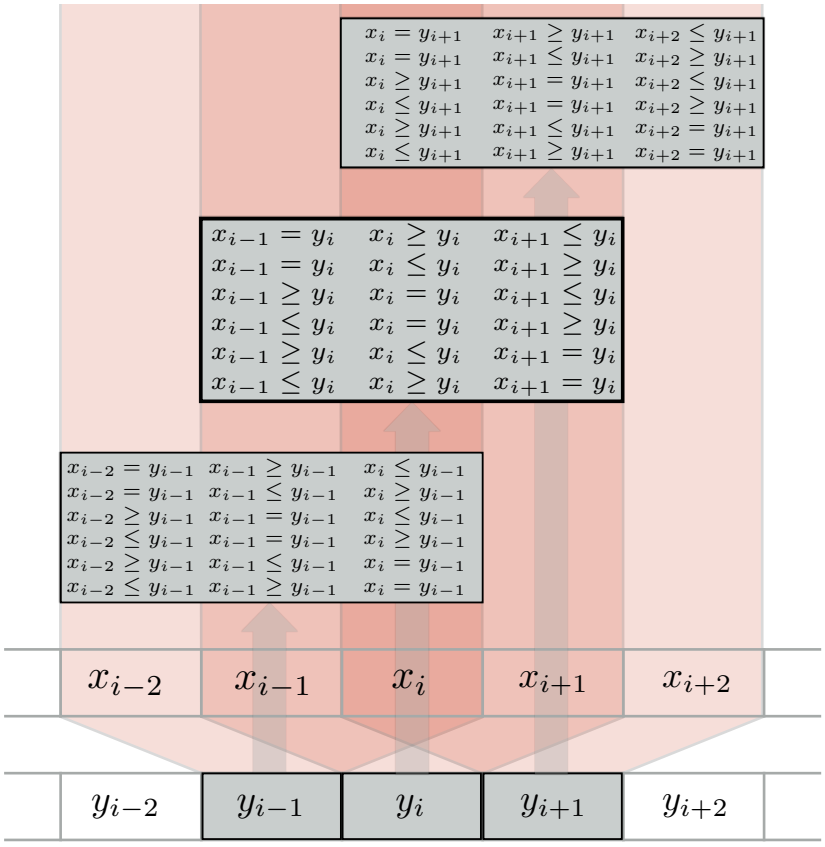


Figure 7.1: Representation of the influence of $\{y_i\}$ on $\{x_i\}$ for the case of $N = 3$.

Property *Let $\{y_i\}$ be the output of a median filter of size N applied to $\{x_i\}$. Then, $\forall i > \lfloor \frac{N}{2} \rfloor$ the following facts hold:*

- *at least one value in X_i is equal to y_i*
- *among the other values of X_i, $\lfloor \frac{N}{2} \rfloor$ of them are equal to or greater than y_i and the remaining $\lfloor \frac{N}{2} \rfloor$ are equal to or lower than y_i.*

As an example, we consider the simple case where $N = 3$. In such case, if $\{y_i\}$ is the sequence to be analyzed and we suppose it is the output of a median filter with size 3 applied to an unknown sequence $\{x_i\}$, a generic value $y_i, i > \lfloor \frac{N}{2} \rfloor$ introduces some knowledge on the 3 elements of X_i, as illustrated in Fig. 7.1. In particular, we can observe that x_{i-1}, x_i and x_{i+1} must satisfy at least one of the 6 sets of conditions reported in the central grey block in Fig. 7.1, and the same holds for all the values in $\{y_i\}$: y_{i+1} enforces 6 possible sets of conditions involving the 3 elements x_i, x_{i+1}, x_{i+2}, as well as y_{i-1} for x_{i-2}, x_{i-1}, x_i, and so on. We can replicate the procedure for a generic filter size N, and we obtain that each y_i affects the N values in X_i by imposing a number of possible sets

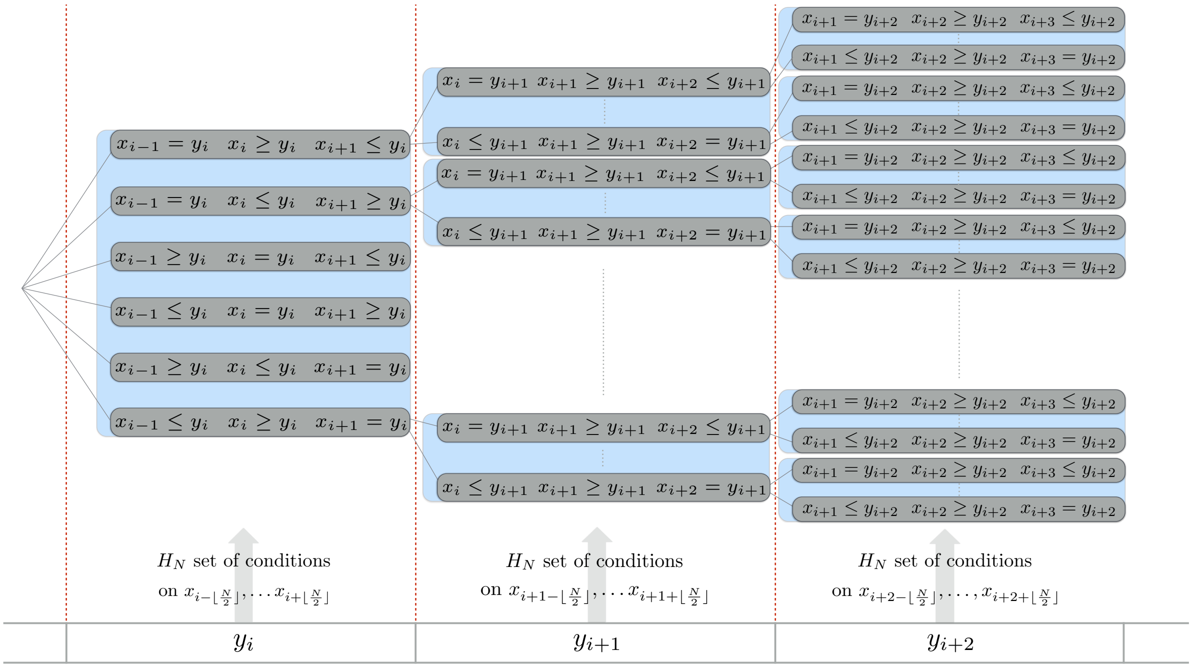


Figure 7.2: Hypothesis tree for a median filter $N = 3$.

of conditions that is equal to

$$H_N = N \frac{(N-1)!}{\left(\frac{N-1}{2}\right)! \left(\frac{N-1}{2}\right)!}.$$

Clearly, H_N significantly increases with N: $H_N = 6, 30, 140, 630$ for $N = 3, 5, 7, 9$, respectively.

If we start from a value y_i and move forward in $\{y_i\}$, at each step we will have H_N possible systems introduced and the elements of $\{x_i\}$ must fulfill at least one of them. This can be represented by a tree, as in Fig. 7.2, where at each node one of the possible set of conditions is added to the ones cumulated in the previous steps along the corresponding path, thus obtaining an equality/inequality system at each node. Now, at each step the systems introduced and the ones of the previous step will share $N-1$ overlapping variables (the intersection of X_i and X_{i+1}). Hence, according to the values in $\{y_i\}$, at each node the system cumulated along the branch might contain conditions on the same variable that are not compatible: in such cases, the cumulated system has an empty feasibility region and we will define the branch as *unfeasible*; otherwise, we will denote it as *feasible*.

Clearly, if at a certain step j all the branches are unfeasible, it means that no sequence $\{x_i\}$ exists such that it could generate $\{y_i\}$ when median filtered with size N. In other words, we have the deterministic proof that $\{y_i\}$ *is not* the output of a median filter of size N and we obtain a response for the forensic problem we face. Although the specific framework and mathematical tools used are different, such approach can be compared to

the ones proposed in [134] and [58], where a similar rationale is employed for the detection of resampling in signals and steganography in digital images, respectively.

7.2.2 Algorithmic checking procedure

In the light of the above, a possible approach for analyzing a given sequence $\{y_i\}$ could be to progressively scan it and determine at each step whether the tree generated by the samples contains at least one feasible branch, meaning that a median filtering with size N might have occurred. In this regard, we propose a recursive algorithmic procedure for the analysis of the sequence $\{y_i\}$ starting from a generic element at position i up to a certain number T_{max} of successive values. It is based on a recursive algorithm consisting of a depth-first visit of the tree and its pseudo-code is reported in Algorithm 1. In particular, it takes as input arguments the size N of the median filter, the sequence $\{y_i\}$, the index i of the first element of $\{y_i\}$ to be considered, the maximum number T_{max} of successive values to be scanned and the current level T of the tree (which is initially set to 1 and increases at each iteration up to T_{max}).

Algorithm 1

```
function CHECK(N,{y_i},i,T,T_max)
    Get y_i from {y_i}
    Create the H_N sets of conditions imposed by y_i
    for each set of conditions do
        Check conditions overlapping for that T
        if conditions are compatible then
            if T = T_max then
                return OK
            else
                R = CHECK(N, {y_i},i + 1,T + 1,T_max)
                if R = OK then
                    return OK
                end if
            end if
        end if
    end for
    return ¬OK
end function
```

In other words, at each call of the CHECK function the algorithm creates all the H_N possible sets of conditions generated by the sample at the current location and check their consistency with the existing branches (created at the previous call) in a sequential order. When the first feasible condition is found, the function recursively launches itself and moves to the successive sample. As we observed, the number H_N substantially increases with N, thus leading to a higher computational complexity of the algorithm when raising N.

Fig. 7.3 shows an example referred to a practical case for the filter size $N = 3$. Here,

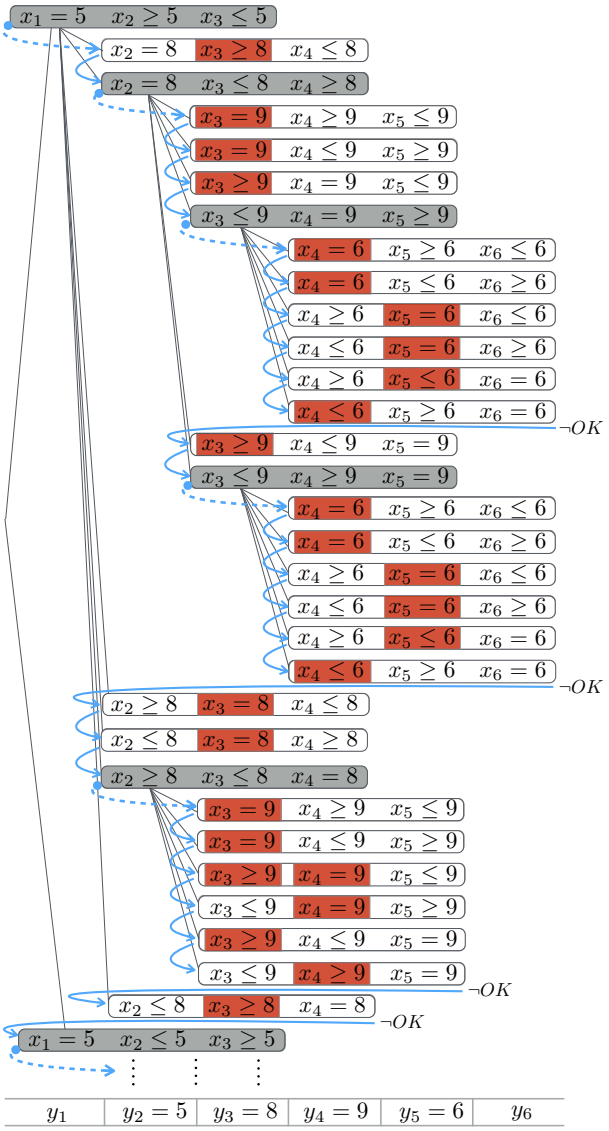


Figure 7.3: Representation of the CHECK function with $N = 3$ for a specific sequence $\{y_i\}$ starting from the position 2.

we consider a sequence $\{y_i\}$ such that

$$y_2 = 5, \quad y_3 = 8, \quad y_4 = 9, \quad y_5 = 6,$$

and illustrate the algorithmic procedure when CHECK(3,$\{y_i\}$,2,1,T_{max}) (where $T_{max} \geq 4$) is called. The analyzed sequence is reported at the bottom and the tree generated is represented above. The blue arrows show the branches sequentially analyzed by the algorithm, where the dotted ones indicate a new call of the CHECK function. Conditions on same elements of $\{x_i\}$ are aligned vertically and a red box indicates that an explicit inconsistency with respect to the previous set of conditions along the current branch is

found. On the other hand, a grey box indicates that the current set of conditions is compatible and the CHECK function is recursively called with the third and fourth input arguments increased by a unit. For the sake of brevity, only the check on the first set of conditions generated by the first sample $y_2 = 5$ is represented (i.e., $x_1 = 5, x_2 \geq 5, x_3 \leq 5$); however, the following ones are treated in the same way and all of them turn out to be closed, thus showing that the analyzed sequence cannot be generated by a median filter of size 3.

Although a progressive scanning by means of such algorithmic procedure would represent a valid solution from a theoretical point of view, the actual application to data sequences of considerable length is computationally demanding, especially when N increases. However, we will see in the next section that we can exploit additional properties of the median filter and adopt smarter strategies in order to simplify the analysis from both a theoretical and a computational perspective.

7.3 Unfeasible classes and $\mathcal{N}$-detectors

A peculiar property of the median filter is the fact that it is based on a sorting operation and, because of that, the set of conditions on $\{x_i\}$ that we defined in the previous section are composed of equalities and non-strict inequalities. By looking at Fig. 7.2, we have that the structure of the tree (the number and kind of conditions added at each step) is determined uniquely by the specific filter size N we are considering. On the other hand, the feasibility of each branch only depends on the values of the y_i's, as it is easy to observe that in an equality/inequality system the existence of a solution only depends on the order relations between all the constant terms involved, that in our case are represented by the elements of $\{y_i\}$. Indeed, in Section 7.2.2 we have shown how to verify that for $N = 3$ a sequence such that

$$y_2 = 5, \quad y_3 = 8, \quad y_4 = 9, \quad y_5 = 6,$$

generates a tree with only unfeasible branches, but we would have obtained the same result by applying the checking procedure to any sequence with the same 6 order relations between y_2, y_3, y_4 and y_5, i.e., such that

$$y_2 < y_3, \quad y_2 < y_4, \quad y_2 < y_5,$$

$$y_3 < y_4, \quad y_3 > y_5,$$

$$y_4 > y_5.$$

Let us now suppose to analyze a generic subsequence of $\{y_i\}$ composed of L elements in consecutive positions, regardless of its starting point in $\{y_i\}$, which can be seen as a vector of length L with values in $\mathbb{R}$. Thus, we can identify $\mathbb{R}^L$ as the set of all such possible objects and, consistently with the notation on trees, we can denote each vector $\mathbf{a}_L \in \mathbb{R}^L$ as *feasible* (*unfeasible*) for a filter size N if the corresponding tree, generated as described in Section 7.2, contains at least one feasible branch (contains only unfeasible branches).

According to the previous observations, we can give the following definition and lemma:

Definition *Let $\sim$ be a binary relation over $\mathbb{R}^L$ such that for $\mathbf{a}_L, \mathbf{b}_L \in \mathbb{R}^L$*

$$\mathbf{a}_L \sim \mathbf{b}_L \iff \begin{array}{l} \textit{all the } L(L-1)/2 \textit{ order relations} \\ \textit{between the elements of } \mathbf{a}_L \\ \textit{hold also for the corresponding} \\ \textit{elements of } \mathbf{b}_L \end{array}$$

Example

$$(1,3,2,4) \sim (100,300,200,400),$$
$$(1,3,2,4) \sim (-20,-1,-7,50),$$
$$(1,3,2,4) \nsim (-50,0,50,100).$$

Lemma *If $\mathbf{a}_L \in \mathbb{R}^L$ is feasible (unfeasible) for a certain filter size N, then any $\mathbf{b}_L \in \mathbb{R}^L$ such that $\mathbf{b}_L \sim \mathbf{a}_L$ is feasible (unfeasible) for N.*

Proof: Let $\mathcal{A}$ and $\mathcal{B}$ be the trees generated for the fixed value of N from $\mathbf{a}_L$ and $\mathbf{b}_L$, respectively, as described in Section 7.2.1 (see Fig. 7.2). Let then $F_{\mathcal{A}}(i,l)$ be the cumulated system at the i-th node of the l-th level of $\mathcal{A}$, where $1 \leq i \leq (H_N)^l$ and $1 \leq l \leq L$; the same holds for $F_{\mathcal{B}}(i,l)$. Being fixed the value of N, $F_{\mathcal{A}}(i,l)$ and $F_{\mathcal{B}}(i,l)$ share the same variables and they are subject to the very same equality or inequality conditions with the exception of the constant terms, which are given by the elements of $\mathbf{a}_L$ and $\mathbf{b}_L$. By hypothesis, the corresponding elements of $\mathbf{a}_L$ and $\mathbf{b}_L$ have the same order relations and the systems $F_{\mathcal{A}}(i,l)$ and $F_{\mathcal{B}}(i,l)$ have either empty or not empty feasibility regions. □

In other words, we can limit the analysis to the classes on $\mathbb{R}^L$ (that we will denote as *L-classes*) defined by the relation $\sim$, since all the vectors of $\mathbb{R}^L$ with the same order relations between their components will be either feasible or unfeasible. This represents a crucial result, as such L-classes are in a finite number (for a given L they can be explicitly determined by means of combinatorics rules) and we can perform our analysis *a priori*, thus avoiding the application of the algorithmic procedure in 7.2.2 to the given sequence. As an example, we proved in Section 7.2.2 that $(5,8,9,6)$ is unfeasible in $\mathbb{R}^4$ for $N=3$. Given a generic sequence $\{y_i\}$, if we find along the sequence 4 values in a consecutive position which have the same order relations as $(5,8,9,6)$, we can deterministically assert that such sequence (or such part of the sequence) has not been median filtered with $N=3$.

To this extent, we can generate all the L-classes for any value of L and determine which of them are feasible or unfeasible for a given filter size N. We will then denote as $\mathcal{U}_L^N$ the set of L-classes that are unfeasible for a certain N and do not contain any L'-class, $L' < L$, that is itself unfeasible for N. By building such sets, we can obtain a set of patterns (meant as order relations between consecutive elements) to be sought in the given sequence $\{y_i\}$ in order to determine whether it has been filtered or not, as it is explored in the next section.

7.3.1 Identification of feasible and unfeasible classes

In order to identify *a priori* the feasible and unfeasible patterns for a certain filter size N, we exploited the algorithm proposed in 7.2.2 by analyzing the L-classes up to a certain

length.

First, we observed that the filter size N determines a maximum length for its unfeasible classes, which is equal to $2N-1$. Indeed, the following lemma holds:

Lemma *Let $\{x_i\}$ be a numerical sequence in $\mathbb{R}$ and $\{y_i\}$ the output of a median filter with size N applied to $\{x_i\}$. Then each y_i will be related to its adjacent values within a maximum window of size $2N-1$.*

Proof: By definition of the median filter, each y_i depends uniquely on X_i. At the same time, X_i affects the values in the set $Y_i = \{y_k,\, k = i - \lfloor \frac{N}{2} \rfloor, \ldots, i + \lfloor \frac{N}{2} \rfloor\}$, thus establishing a mutual relationship between y_i and the elements of Y_i, whose cardinality is $2N-1$. □

Hence, we have that the maximum length is given by 5, 9, 13 and 17 for $N = 3, 5, 7, 9$, respectively.

Then, we employed the function UnfeasibleClasses, whose pseudocode is reported in Algorithm 2, to identify the unfeasible L-classes for a specific filter size N and for L starting from 2 up to $2N-1$. In particular, at each step all the possible L-classes for the current L are created, the ones which contain a $(L-1)$-class identified as unfeasible at the previous iteration are discarded and, among the remaining ones, the Check function determines which ones are unfeasible, thus obtaining the set $\mathcal{U}_L^N$ for each length L.

Algorithm 2 Identification of unfeasible classes

```
function UnfeasibleClasses(N)
    for L = 2, ..., 2N − 1 do
        Create all the possible L-classes
        for each L-class do
            if it does not contain an element of U_{L−1}^N then
                Choose an a_L ∈ R^L in the current class
                Create the sequence {a_i},
                whose first L values are the elements of a_L
                if Check(N,{a_i},1,1,L)= ¬OK then
                    Save current class in U_L^N
                end if
            end if
        end for
    end for
end function
```

In Table 7.1, we report the cardinality of the $\mathcal{U}_L^N$ obtained, indicated as $|\mathcal{U}_L^N|$. The computational complexity of the procedure increases both with L (because of the total number of L-classes, which is reported in the second column) and with N (because of the number of possible sets of conditions). Thus, we limited the analysis to the classes up to $L = 11$ and in the hardest cases (marked with the symbol $*$), instead of all the possible L-classes, we considered only the ones where the relations between the L elements are strict inequalities, which can be seen as the $L!$ possible permutations of L elements (for $L = 11$ we have almost 40 million permutations). Although this does not provide an

Table 7.1: Number of unfeasible L-classes derived for different values of N. The symbol $*$ means that only classes with strict inequality relations between the L elements have been considered. The last row reports the total time (in seconds) necessary to derive all the unfeasible classes of the corresponding value of N.

L	N. of L-classes	$\lvert\mathcal{U}_L^3\rvert$	$\lvert\mathcal{U}_L^5\rvert$	$\lvert\mathcal{U}_L^7\rvert$	$\lvert\mathcal{U}_L^9\rvert$
2	2	0	0	0	0
3	13	0	0	0	0
4	75	12	0	0	0
5	541	20	60	36	36
6	4683	0	468	270	222
7	47293	0	74	1712	980
8	545835	0	34	7666	5578
9	7087261	0	2	1802	31496
10	102247563	0	0	838	29776*
11	$\sim$ 1 billion	0	0	478	6510*
Total computation time		0.15s	580.3s	27320s	51960s

exhaustive analysis ($2N-1$ would be higher for $N=7,9$), the number of unfeasible classes detected among the $L!$ considered is significant and sufficient to correctly detect the filter, as we will see in the next sections. It is to be pointed out that, although it is time consuming for higher values of N, the above operation can be performed off-line and once-for-all, and can be easily parallelized; moreover, aggregating all unfeasible classes detected among different values of L and N requires very limited memory requirements.

As an example, in Fig. 7.4 the 12 unfeasible 4-classes for $N=3$ are graphically represented, where the distance between each value has been normalized to a common unit. We can notice that in most of the classes the values satisfy strict inequalities, while in 4 classes the first and the last elements are equal. Moreover, for each class its symmetric counterpart in the vertical and horizontal direction are included in $\mathcal{U}_4^3$. Indeed, it is easy to observe that in case of vertical or horizontal symmetry the building of the tree leads to the same results, thus suggesting that a further equivalence could be introduced in order to represent the unfeasible patterns in a more compact way.

Furthermore, it is interesting to visualize how much the unfeasible classes are similar

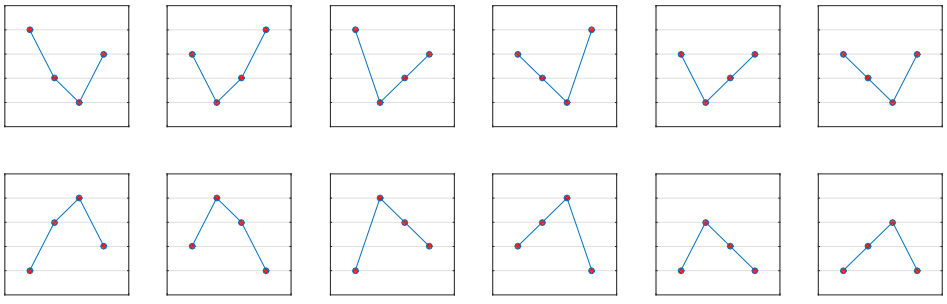


Figure 7.4: Graphical visualization of the elements in $\mathcal{U}_4^3$.

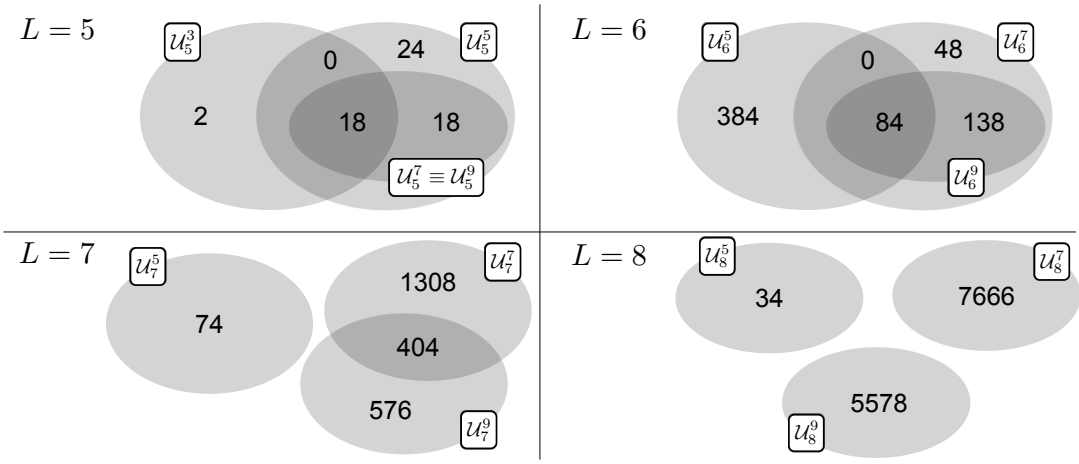
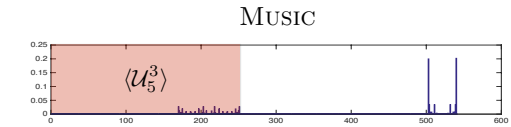


Figure 7.5: Relative sharing of the unfeasible classes among different values of N and L. A number indicating the cardinality is placed in every set. Notice that for $N = 3$ the sets $\mathcal{U}_4^N$ and $\mathcal{U}_5^N$ are empty.

for different values of N. To this end, we identified the unfeasible classes at a certain L that are shared by more than one value of N and the ones that are unfeasible for only one value of N. In Fig. 7.5, such analysis is graphically represented as Euler diagram, highlighting the different behaviour of the unfeasible classes when L varies. For instance, we can notice that the sets $\mathcal{U}_5^7$ and $\mathcal{U}_5^9$ coincide and, in general, the unfeasible classes for $N = 7$ and $N = 9$ have a significant overlap (which decreases by increasing L). Clearly, this intrinsically affects the detection performance of the method when a different window size is used in the filtering. Indeed, it is easy to predict that if a sequence has been median filtered with $N = 7$, the probability that it will contain classes that are unfeasible for $N = 9$ will strongly decrease, as most of them are unfeasible also for $N = 7$ and have been certainly removed.

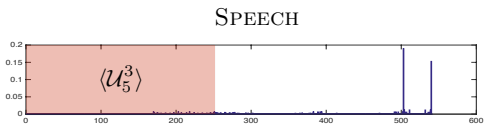
7.3.2 $\mathcal{N}$-detectors

Once the sets $\mathcal{U}_L^N$ are identified, the definition of simple and fast detectors is straightforward. Indeed, differently as the solution proposed in Section 7.2.2, it is now possible to simply scan the sequence progressively and check whether the elements of $\{y_i\}$ fulfill or not the order relations corresponding to the classes in the sets $\mathcal{U}_L^N$: if such classes are present in the sequence, we can deterministically classify $\{y_i\}$ as not filtered; if none of the unfeasible classes is found, we classify the sequence as filtered. Although this second assumption is not deterministic (unfeasible patterns might be missing also in a not filtered sequence), we will see that in practice the false alarm rate is basically null for sequences with a sufficient length, since pristine signals most likely contain unfeasible classes. A deeper analysis of such aspect is provided in Section 7.4.1, where we establish a relationship between the length of the analyzed sequence and the false alarm probability. Moreover, the implementation of such procedure leads to algorithms with a quite low computational complexity, consisting of a simple check on order relations.



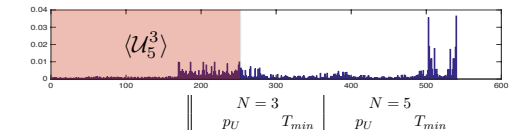
MUSIC

	$N=3$		$N=5$	
	p_U	T_{min}	p_U	T_{min}
ORIGINAL	0.40	50	0.32	84
MEDIAN 3	0	-	0.23	126
MEDIAN 5	$9.8 \cdot 10^{-2}$	2235	0	-
MEDIAN 7	$1.1 \cdot 10^{-2}$	2025	$5.1 \cdot 10^{-3}$	6307
MEDIAN 9	$1.1 \cdot 10^{-2}$	1180	$1.3 \cdot 10^{-2}$	2317



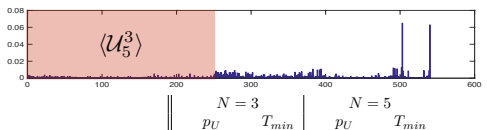
SPEECH

	$N=3$		$N=5$	
	p_U	T_{min}	p_U	T_{min}
ORIGINAL	0.47	40	0.39	70
MEDIAN 3	0	-	0.19	147
MEDIAN 5	$2.0 \cdot 10^{-2}$	1100	0	-
MEDIAN 7	$1.4 \cdot 10^{-2}$	1575	$1.5 \cdot 10^{-2}$	2037
MEDIAN 9	$1.2 \cdot 10^{-2}$	1825	$1.7 \cdot 10^{-2}$	1813



STOCK

	$N=3$		$N=5$	
	p_U	T_{min}	p_U	T_{min}
ORIGINAL	0.40	45	0.38	70
MEDIAN 3	0	-	0.12	252
MEDIAN 5	$1.7 \cdot 10^{-2}$	1315	0	-
MEDIAN 7	$1.0 \cdot 10^{-2}$	2110	$9.4 \cdot 10^{-3}$	3430
MEDIAN 9	$8.1 \cdot 10^{-3}$	2845	$7.8 \cdot 10^{-3}$	4144



ECG

	$N=3$		$N=5$	
	p_U	T_{min}	p_U	T_{min}
ORIGINAL	0.20	100	0.21	133
MEDIAN 3	0	-	$5.2 \cdot 10^{-2}$	602
MEDIAN 5	$5.3 \cdot 10^{-3}$	4300	0	-
MEDIAN 7	$6.0 \cdot 10^{-3}$	3810	$6.0 \cdot 10^{-3}$	5383
MEDIAN 9	$3.1 \cdot 10^{-3}$	7300	$4.0 \cdot 10^{-3}$	7959

Figure 7.6: Pmf of the 5-classes and 7-classes in the different datasets. The histogram depicted refers to the ORIGINAL case for each dataset and the 252 classes belonging to $\langle \mathcal{U}_5^3 \rangle$ have been placed in the first bins. For the different processing (reported row wise), we computed the value of p_U and the value T_{min}, by imposing a false alarm probability upper bound equal to 0.01.

Finally, the knowledge of the $\mathcal{U}_L^N$'s for the different values of N allows for a median filter detection targeted to one or more values of the filter size N. Precisely, considering a generic set of integer odd values $\mathcal{N}$, we can define its corresponding $\mathcal{N}$-detector: in this case, the presence of all the unfeasible classes in $\bigcup_{N \in \mathcal{N}} \mathcal{U}_L^N$ will be checked and the sequence will be considered as positive to median filtering detection if only feasible classes for at least one value of $N \in \mathcal{N}$ are detected; on the other hand, we consider the sequence as negative to median filtering detection if at least one unfeasible class for each value of $N \in \mathcal{N}$ is detected.

7.4 Experimental results

The proposed detection method has been tested on various kinds of 1D data. In particular, we considered 4 different datasets:

- **Music**: music audio clips of different length and sources have been taken from the entire version of the publicly available dataset used in [129], which includes 64 clips of 30 seconds and a genre collection composed by 1000 tracks of 30 seconds. They are stored in .wav and .au format and their sources vary from CD to radio and microphone recordings. The total number of clips is 1064, each one sampled at 22050Hz and consisting of 661500 samples.

- **Speech**: similarly, speech audio clips have been taken from [129] (where the technical specifications are the same as the music clips) and from the AMI Corpus [3], which includes 36 conference recordings of 20-60 minutes. From the latter, the first 20

millions samples (20 minutes) of each recording have been considered. The total duration of the dataset is 752 minutes.

- **Stock**: we downloaded historical stock data [2] of all the companies listed on NASDAQ stock exchange since at least 10 years. In particular, we considered the daily closing price of each company's security, thus obtaining a total number of 1299 sequences whose length varies from 3650 to about 9000 (depending on the date they entered the stock market).
- **Ecg**: Electrocardiogram (ECG) data (used in [99]) have been downloaded from the publicly available MIT-BIH Arrhythmia Database [93], the MIT-BIH Supraventricular Arrhythmia Database [64] and from the European ST-T Database [126]. The three datasets provide a total number of 216 ECG sequences containing on average 300000 samples.

7.4.1 False alarm probability analysis

As previously stressed, our detection method can guarantee a null rate of false negatives (median filtered data classified as not filtered), as filtered sequences will not present unfeasible classes by definition. On the other hand, not filtered sequences that do not contain unfeasible classes are possible in principle. In those cases, the detector would fail and classify the sequence as filtered, thus leading to false positives.

In order to quantify such false alarm probability, we performed a preliminary analysis on a subset of each dataset. Let us consider a value of N and a sequence with length equal to T. Then, we want to estimate $p_{FA}(T)$, the false alarm probability of the $\{N\}$-detector for a given dataset as a function of T, which can be seen as the probability of finding only feasible classes for N in sequences with T samples that have not been filtered. For the sake of simplicity, let L be such that $\exists L' \leq L$ with $\mathcal{U}_{L'}^N \neq \emptyset$, and let us suppose to consider only non-overlapping subsequences of length L contained in the dataset, whose corresponding L-class is represented by a random variable C_L.

Then, we can assume that all the L-classes (meant as the L-classes corresponding to the subsequences of length L) contained in the dataset are independent realizations of C_L, and that we have an estimate of the probability mass function of C_L. Now, the sample space of C_L coincides with all the possible L-classes and is composed of two disjoint parts: the L-classes that do not contain nor are themselves unfeasible classes and the ones that either belong to $\mathcal{U}_L^N$ or contain a L'-class that belongs to $\mathcal{U}_{L'}^N$. We will denote the latter set as $\langle \mathcal{U}_L^N \rangle$ and by summing up the pmf values of all the elements in $\langle \mathcal{U}_L^N \rangle$ we can quantify the probability p_U that a realization of C_L is unfeasible.

Finally, these probability values provide some information on $p_{FA}(T)$ for a generic sequence. Indeed, a sequence of length T contains $\lfloor T/L \rfloor$ non-overlapping subsequences of length L and, thanks to the independence assumption and the knowledge of p_U, we can compute the probability of finding a given number of L-classes in $\langle \mathcal{U}_L^N \rangle$ among them. This is done be seeing the sequential analysis of the non-overlapping subsequences as a Bernoulli process (where the success is represented by the fact that the corresponding L-class belongs to $\langle \mathcal{U}_L^N \rangle$) and model the number of successes as a binomial distribution with parameters $\lfloor T/L \rfloor$ and p_U. We can then obtain the probability $p_{FA}^L(T)$ of finding

only feasible L-classes among the $\lfloor T/L \rfloor$ non-overlapping subsequences in the sequence as follows:

$$p_{FA}(T) \leq p_{FA}^{L}(T) = \binom{\lfloor T/L \rfloor}{0} p_U^0 (1 - p_U)^{\lfloor T/L \rfloor} = (1 - p_U)^{\lfloor T/L \rfloor}. \tag{7.1}$$

As stated in (7.1), $p_{FA}^{L}(T)$ is an upper bound of the global false alarm probability $p_{FA}(T)$ as it refers to non-overlapping subsequences only, while overlapping ones could belong to $\langle \mathcal{U}_L^N \rangle$ as well; moreover, unfeasible classes for N with a higher length might be present in the sequence. However, such a result allows us to find a value of T for which the false alarm probability is certainly lower than a desired value.

In Fig. 7.6, we report the results of such kind of analysis for two pair of N and L values: for $N = 3$ we considered $L = 5$ (which excludes the possibility of longer unfeasible sequences) and for $N = 5$ we considered $L = 7$, as higher values would imply a significantly higher computational cost. For each sequence of each dataset we considered its original version (denoted as ORIGINAL) and we created four other sequences, resulting from the application of a median filter with a varying window size M, respectively denoted as MEDIAN M for $M = 3, 5, 7, 9$. Then, we estimated the pmf in each case by randomly collecting 500000 subsequences of the chosen length and creating a normalized histogram of the corresponding possible L-classes (see Fig. 7.6, where the pmf estimation for the ORIGINAL sequences in the case $N = 3$ and $L = 5$ is represented for each dataset).

By knowing $\mathcal{U}_5^3$ and $\mathcal{U}_7^5$, we identified the 252 5-classes in $\langle \mathcal{U}_5^3 \rangle$ and the 33232 7-classes in $\langle \mathcal{U}_7^5 \rangle$. Then, we computed the value of p_U in each case and, by means of expression (7.1), we derived the length value T_{min} which is necessary to guarantee that $p_{FA}(T) \leq 0.01$. It is interesting to notice that, although the pmfs present some differences among the datasets due to the different nature of the data, the general shape of the histogram is quite similar. We also observe in any case a decrease of p_U when a window size $M = 5, 7, 9$ is used, while it is clearly null when $M = N$. However, the value of T_{min} turns out to be quite low (not higher than 8000 in each case) for both the values of N considered, and allows for acceptable results in terms of detection, as it will be explored in the following experiments.

7.4.2 Filter detection

After assessing the false alarm probability, we applied the $\mathcal{N}$-detectors (as described in Section 7.3.2) for different $\mathcal{N}$ to the sequences of all the datasets (assuming to stop the search as soon as one unfeasible class for each element of $\mathcal{N}$ is found) and tested also the effectiveness of the proposed approach in identifying and discriminating median filtering with respect to other processing. In fact, in addition to the ORIGINAL and MEDIAN M cases, for each sequence of each dataset we also created other two versions, resulting from the application of a moving average filter (with window size 3) and a Gaussian lowpass filter (with window size 3 and standard deviation equal to 0.5), respectively denoted as MOVING AVERAGE and GAUSSIAN LOWPASS.

First, we applied a $\{3, 5, 7, 9\}$-detector, thus using all the unfeasible classes that we

derived in Section 7.3.1. As we specified in Section 7.3.2, the algorithm checks the presence of unfeasible classes for $N = 3, 5, 7, 9$ and the sequence is classified as positive if only feasible classes for at least one value of N are detected, while it is classified as negative if at least one unfeasible class for each value of N is found. Clearly, the ORIGINAL, MOVING AVERAGE and GAUSSIAN LOWPASS cases should be negative to such detector, while the MEDIAN M cases should be positive. As expected from the theory, the rate of false negatives was null in all tests, so in Table 7.2 we report the results only on sequences that were not median filtered in terms of false alarm, which is the only type of error that might occur. By observing the results, we can notice that we also have a null rate of false positives both in the ORIGINAL row (as we predicted in Section 7.4.1) and the MOVING AVERAGE/GAUSSIAN LOWPASS rows, thus showing the ability of the method in distinguishing between median filtering and other kind of processing.

Regarding the complexity and computational time of the detection, the search will be more demanding as N increases, as the number and the length of unfeasible classes increase as well (for $N = 9$, a total number of 74598 unfeasible classes need to be checked with a length up to 11), but the first unfeasible class is usually detected very soon. In Table 7.3, we report the average computational time in seconds necessary to process 100 samples, showing the short time frame for the analysis of not filtered sequences (about 6 milliseconds for a 30 second long audio track).

As further analysis, we applied the detectors for a specific N to sequences that have been median filtered with a window size $M \neq N$, thus evaluating the ability of the technique to discriminate the size of the filter used. For instance, sequences filtered with $M = 3$ should be positive to a $\{3\}$-detector and negative to the $\{5\}$-, $\{7\}$- and $\{9\}$-detectors, and so on. However, we observed in Section 7.3.1 that the sets of unfeasible classes for N and M usually share a number of classes (in particular for $L = 5, 6$), which are certainly removed when a median filter with size M is applied. In addition, the action of a median filter with size $M \neq N$ generally decreases the frequency of occurrence of unfeasible classes for N even though they are feasible for M, as we observed in Section 7.4.1 for $N = 3$. Because of that, the presence of unfeasible classes for N in sequences filtered with size M is less probable, thus affecting the performance of $\{N\}$-detectors. In Tables 7.4, we report the percentage of sequences classified as positives in the different cases. Similarly as before, the different values of M are reported row wise, while the values of N are placed column wise. We can observe that the false alarm rate is generally lower than 1% for the $\{3\}$-detector, which is again coherent with the analysis performed in Section 7.4.1 as the length of the sequence is in any case higher than the correspondent T_{min} values reported in Fig. 7.6. While the false alarm rate is generally acceptable up to $N = 7$, it substantially increases for the $\{9\}$-detector when a median filter with size $M = 5, 7$ is used, thus confirming that such filters tend to remove the unfeasible classes for $N = 9$. However, it is worth pointing out that for the sequences with a substantially higher length (i.e., part of the **Speech** dataset), such classes are generally found.

7.4.3 Comparison with state-of-the-art techniques

We also tested our method against some existing detection techniques. As we stressed, no specific methods for 1D data exists, thus we adapted three different state-of-the-art

	Music	**Speech**	**Stock**	**Ecg**
Original	0%	0%	0%	0%
Moving average	0%	0%	0%	0%
Gaussian lowpass	0%	0%	0%	0%

Table 7.2: False alarm of the $\{3, 5, 7, 9\}$-detector.

	Music	**Speech**	**Stock**	**Ecg**
Original	$8.5 \cdot 10^{-6}$	$8.0 \cdot 10^{-6}$	$3.4 \cdot 10^{-4}$	$1.8 \cdot 10^{-5}$
Moving average	$8.3 \cdot 10^{-6}$	$7.9 \cdot 10^{-6}$	$3.5 \cdot 10^{-4}$	$1.7 \cdot 10^{-5}$
Gaussian lowpass	$8.4 \cdot 10^{-6}$	$7.9 \cdot 10^{-6}$	$3.4 \cdot 10^{-3}$	$1.7 \cdot 10^{-5}$

Table 7.3: Average computational time in seconds of the $\{3, 5, 7, 9\}$-detector for 100 samples.

(a) **Music**

	$N = 3$	$N = 5$	$N = 7$	$N = 9$
Median 3	100%	0%	0.%	0%
Median 5	0%	100%	0.4%	100%
Median 7	0%	0%	100%	100%
Median 9	0%	0%	0.2%	100%

(b) **Speech**

	$N = 3$	$N = 5$	$N = 7$	$N = 9$
Median 3	100%	0%	7.0%	85.3%
Median 5	0%	100%	1.0%	78.0%
Median 7	0%	0%	100%	78.0%
Median 9	0%	0%	0%	100%

(c) **Stock**

	$N = 3$	$N = 5$	$N = 7$	$N = 9$
Median 3	100%	0.3%	0.1%	0%
Median 5	0.4%	100%	6.8%	100%
Median 7	0.3%	16.9%	100%	100%
Median 9	0.7%	25.7%	45.7%	0%

(d) **Ecg**

	$N = 3$	$N = 5$	$N = 7$	$N = 9$
Median 3	100%	0%	0%%	0%
Median 5	0.9%	100%	20.8%	100%
Median 7	0%	13.4%	100%	100%
Median 9	0.1%	11.5%	27.8%	0%

Table 7.4: Percentage of positives for the $\{N\}$-detectors when a different window size is used in the filtering.

approaches that were originally conceived for images (2D median filtering), but their rationale can be applied also to 1D data. In particular, we considered the techniques proposed in [77], [25] and [71]. All of them require some training phase, as the former two are threshold-based and the latter employs an SVM classifier. Thus, we divided each dataset in two equal parts, one used for the training and the other one for the testing phase. For the methods in [77] and [25], the optimal threshold was determined by fixing a maximum false alarm rate of 5% and choosing the threshold value yielding the lowest false

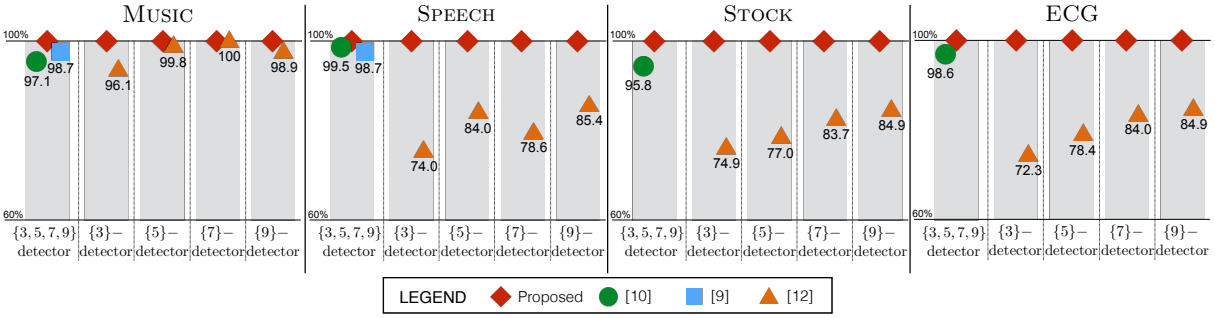


Figure 7.7: Accuracies obtained by applying our approach and the existing state-of-the-art methods.

	Music	**Speech**	**Stock**	**Ecg**
[25]	$7.4 \cdot 10^{-3}$	$6.5 \cdot 10^{-3}$	$5.9 \cdot 10^{-3}$	$5.8 \cdot 10^{-3}$
[77]	$3.4 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$5.8 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$
[71]	$6.2 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$

Table 7.5: Average computational time in seconds of different methods for 100 samples.

negative rate, while for the method in [71] we trained the SVM classifiers as suggested in the original paper.

Because of the different properties of each method, we could perform a comparison only for certain experimental scenarios. Indeed, the detectors [77] and [25] are not targeted to a specific filter size but they indicate that a generic median filter operation has been applied, thus they have been compared with the $\{3, 5, 7, 9\}$-detector. On the other hand, the technique in [71] is able to discriminate among different values of N and it has been compared with different $\{N\}$-detectors. Moreover, the detector in [77] can be applied only on data containing integer values, as it is based on an histogram bin ratio, thus in our experiments we could employ it only for the audio datasets. According to this considerations, in Fig. 7.7 we report the results obtained when applying the different detectors on non median filtered sequences (including original, average filtered, Gaussian lowpass filtered) and median filtered with the same window as the detector. Here, we report the accuracy value (meant as the percentage of sequences correctly classified as filtered or not filtered) in the different cases, as false negatives are also possible for the other methods. In this setting, we can observe that our technique achieves the maximum accuracy in any case. The performance of [25] and [77] is also good in the corresponding scenario, while we can notice that the technique in [71] has a different behaviour throughout the different datasets. Moreover, in Table 7.5 we report the average computational time that is necessary to each technique to process (i.e., extracting the features) 100 samples from the different datasets.

7.4.4 Tampering localization

In this section, we exploit the proposed detection algorithm to locate parts of the 1D data that have been median filtered. In particular, we partition each sequence into seven non-

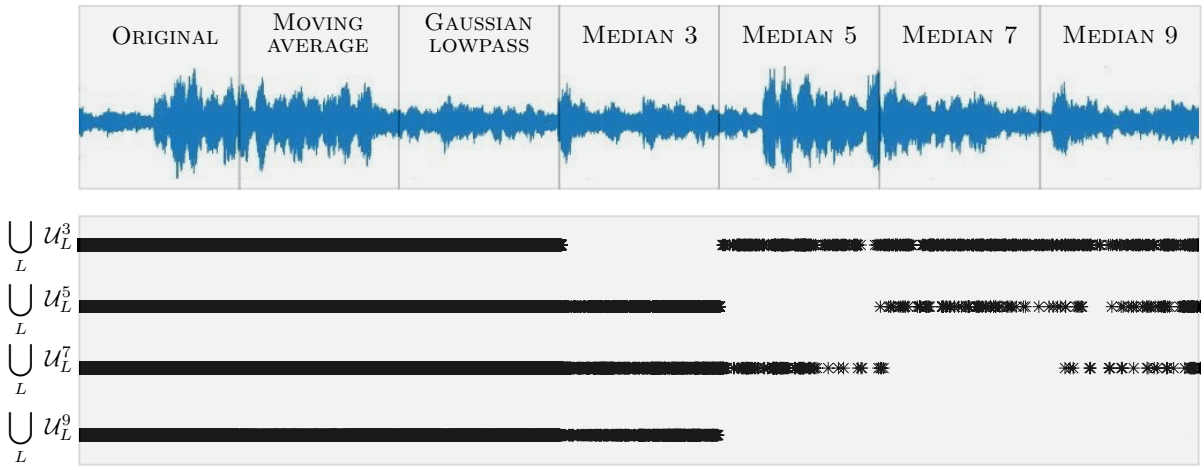


Figure 7.8: Example of median filter localization with the $\{3, 5, 7, 9\}$-detector. The black asterisks are the unfeasible classes detected through the whole 1D signal.

overlapping parts, each of them processed according to the different operations employed in the previous experiments (Original, Moving Average, Gaussian Lowpass, Median with window size $3, 5, 7, 9$, applied in this order). In this phase, we limited the analysis to a subset of the **Music** and **Speech** dataset and applied each processing operation to a signal part of 30000 samples.

We then run the $\{3, 5, 7, 9\}$-detector through the whole sequence (i.e., not stopping at the first unfeasible class detected but analyzing the sequence entirely) and perform a local analysis by considering smaller non-overlapping blocks of 5000 samples. In particular, we classify each of them as positive to median filtering if only feasible classes for at least one values of N lie within it (i.e., blocks belonging to the first three parts are true negatives and the ones belonging to the last four parts are true positives). In Fig. 7.8, we graphically represent the behaviour of the detector by means of an exemplifying test sequence. The black asterisks below the signal are the unfeasible classes detected through the whole sequence for the different values of N; clearly, no unfeasible classes for a certain N are detected in the part of the sequence median filtered with window size N. Also in this case, we obtain 0% false negatives in every sequence, which means that at least one unfeasible class is detected in each one of the not median filtered blocks. Moreover, we can notice that the unfeasible classes for any N are dense in the first three parts, while they are more sparse when another filter size is used. In particular, coherently with the results in the previous section, no unfeasible classes for $N = 9$ are found in the last three parts, although this does not affect the performance of the $\{3, 5, 7, 9\}$-detector.

7.4.5 Robustness analysis

In this section, we deal with the issues of assessing the robustness of our method with respect to post processing, i.e., we consider the problem of detecting median filtering even when a further successive operation is applied after it. Firstly, we can observe that our

technique is based on the order relationships between the samples and it is robust to any post-processing that preserve them, such as amplitude re-scaling, monotonic corrections, normalization, shifting. With regards to other operations, it is easy to state that when the median filter has not been the very last process, the method loses its deterministic nature since the post processing would potentially introduce unfeasible classes. However, we can notice that in such situation the effect of the previous median filtering is still visible in the distribution of the unfeasible classes detected in the sequence. In particular, we considered the sequences of the different datasets with no processing at all (denoted again as ORIGINAL) and we processed the same sequences by first applying a median filter with $N = 3$ or $N = 5$ followed by a post processing operation among moving average and Gaussian lowpass filter, thus obtaining four different scenarios. Then, we run the respective $\{N\}$-detector on all the sequences and we created an histogram of all the unfeasible classes for that value of N, normalized by the total number of unfeasible classes detected. In particular, for $N = 3$ the histogram has $|\mathcal{U}_4^3| + |\mathcal{U}_5^3| = 32$ bins and for $N = 5$ it has $|\mathcal{U}_5^5| + |\mathcal{U}_6^5| + |\mathcal{U}_7^5| + |\mathcal{U}_8^5| + |\mathcal{U}_9^5| = 638$ bins.

A possible approach is to use the histogram bin values as features and feed a classifier. Hence, as we did for the state-of-the-art approaches, we divided each dataset in two equal parts, one for training and one for testing, and we extracted features from each training set. Then, for each of the four scenarios we trained an SVM classifier with Gaussian kernel (optimal parameters have been derived by means of a grid search procedure) using features from all the datasets. Finally, we performed a classification for the testing set of each dataset in the different scenarios. In Table 7.6, we report the accuracies obtained. We can observe that the accuracy values are higher than 90% almost in any case and, as expected, the performance are better for the case of $N = 5$ and the Gaussian lowpass filter, which has less impact than the moving average filter. In order to have a comparison with state-of-the-art methods, we considered the approach in [71] (the only one which detects median filter with a specific value of N) and report in Table 7.7 the results obtained by replicating the same experimental settings, showing a clear performance drop in case of post-processing and worse accuracies with respect to the proposed method.

	Music	**Speech**	**Stock**	**Ecg**
ORIGINAL vs MEDIAN $N = 3$ + MOVING AVERAGE	97.8%	93.7%	92.2%	73.1%
ORIGINAL vs MEDIAN $N = 5$ + MOVING AVERAGE	100%	100%	98.6%	94.4%
ORIGINAL vs MEDIAN $N = 3$ + GAUSSIAN LOWPASS	99.0%	96.8%	96.8%	78.2%
ORIGINAL vs MEDIAN $N = 5$ + GAUSSIAN LOWPASS	100%	100%	97.8%	98.2%

Table 7.6: Classification accuracy by means of SVM with post-processing.

	Music	**Speech**	**Stock**	**Ecg**
ORIGINAL vs MEDIAN $N = 3$ + MOVING AVERAGE	55.0%	48.4%	50.7%	50.1%
ORIGINAL vs MEDIAN $N = 5$ + MOVING AVERAGE	50.0%	56.2%	74.4%	55.4%
ORIGINAL vs MEDIAN $N = 3$ + GAUSSIAN LOWPASS	62.3%	67.2%	56.7%	60.2%
ORIGINAL vs MEDIAN $N = 5$ + GAUSSIAN LOWPASS	66.2%	76.6%	69.4%	74.2%

Table 7.7: Classification accuracy by means of method in [71] with post-processing.

7.5 Discussion

We have proposed a forensic detector of median filtering on 1D data based on deterministic properties of such processing operation. According to a well defined theoretical rationale, a set of patterns that cannot be present in median filtered sequences have been computed offline and the final algorithm consists in searching such patterns in the test sequence. The proposed method has been tested on 1D signals and time series coming from different sources, and proved to be extremely accurate in detecting and locating the occurrence of median filtering as well as identifying the size of the filter employed, which is a quite rare feature in existing techniques for 2D median forensics. Moreover, we also proved that the study of the unfeasible classes can be used to detect median filtering also in case of post-processing.

Such promising results open the way for future developments of this work in different directions. A natural step further would be to approach 2D median filtering, which is commonly applied to images. Unfortunately, treating bidimensional data and filters introduces significant problems, both in terms of theoretical results and computational complexity. Indeed, although the theoretical concepts can be easily extended, the derivation of the unfeasible classes presents two main issues: due to the distribution in the 2D domain, the actual overlapping variables at each step is reduced with respect to the 1D case (at most 6 overlapping variables for a 3×3 filter), together with the chance of encountering unfeasible branches; as a consequence, the number of possible branches in the tree (630 new ones are introduced at each step for a 3×3 filter) and the number of L-classes required is higher, leading to extremely demanding computational efforts. For this reasons, a significant optimization of the technique would be required and will be certainly subject of future work.

Moreover, much work can be developed regarding a further analysis of robustness issues of the proposed method, by designing more specific and advanced approaches allowing for the identification of the median filtering occurrence even after a wider range of post-processing operations.

Finally, whether the study of deterministic properties can be extended also to other kind of processing is still an open and fascinating research question, that will be certainly subject to future work.

Chapter 8

Conclusion

In this doctoral study we have developed innovative methodologies for the forensic analysis of multimedia data. For each of the solutions proposed, experimental validation on benchmarking datasets has been carried out, thus identifying the strengths and contributions with respect to existing tools, as well as aspects leaving space of improvement. The latter have been discussed at the conclusion of each chapter, being natural future research directions. However, we can identify two general aspects that particularly deserve our attention.

First, we stress that the main rationale behind our work has been to exploit statistical and deterministic properties that are known and common to multimedia data. When possible, we have tried to derive closed-form models resulting in well-defined discrimination tests that are generally computationally lightweight and easy to convey, particularly relevant aspects in digital forensic practices. With this respect, it is certainly possible to devise further advances, both for statistical and deterministic models, that could include the use of theoretical tools from statistics, information theory, detection theory. For instance, assessing the fundamental limits of our detectors and determine under which conditions the hypotheses we consider are actually distinguishable would represent a significant step forward.

Second, it is worth pointing out that the experimental validation of our novel approaches has been performed in controlled scenarios with the goal of assessing the performance and foundations of the methods, as it happens for the vast majority of existing multimedia forensic techniques. However, real-world cases include an extremely wide variety of contents, sources, resolution, processing history. A future challenge for this populated group of methodologies is to go beyond laboratory hypotheses, which are not necessarily met in data coming from unreliable sources [140]. For instance, this issue arises when assessing the reliability of data found in the web, both for content verification and investigation practices. With this respect, on one hand the proposed solutions have the positive features to be largely independent from the semantic content of the objects under investigation and to require none or very limited training phases on preliminary datasets. Thus, they can be considered as *off-the-shelf* tools to analyze data with diverse content and coming from diverse sources. On the other hand, opening to multimedia web applications (social networks are the perfect examples) implies also a wider variety of pre- and post- processing operations, which can compromise the effectiveness of the existing

techniques. To this extent, a promising possibility (explored in recent works [104, 6]), would be to support the forensic analysis by exploiting the richness of the web in terms of metadata and visual/textual information.

In conclusion, extensions of the contributions proposed in this thesis can be devised both from a theoretical and application-driven perspective. How to jointly cope with such aspects and design comprehensive solutions in these directions represent the main future challenge for our work.

Acknowledgements

It was almost four years ago when I decided to take my personal "leap of faith" and start this PhD. While it is hard to realize how many things have changed since then, it is very easy to think of people who strongly deserve a thought of gratitude at the end of the road.
My first heartfelt thanks go to my advisor Giulia Boato for the many ways she made the difference during my PhD. Starting from the decision of giving me a chance in the first place, somehow taking her own leap of faith. During these years, she constantly provided me with precious scientific guidance and support, giving me valuable feedbacks, granting me the best opportunities and always creating a positive working environment. Her attitude and ability as advisor to turn my many concerns into challenges to be taken up has been priceless to me, and had countless positive effects on my research and my daily work life. For this, and many other things, I owe her my most sincere gratitude.

Then, I really want to thank Fernando Pèrez-Gonzàlez, a very important person for the research path of my PhD. Having the chance to collaborate with him and spend a visiting period within his group at University of Vigo has been an incredibly rewarding life and work experience, that truly represented an added value to my PhD. Special thanks also go to Pedro Comesaña-Alfaro for the interesting collaboration we had and his scientific support during my stay in Vigo.

I also want to thank Rainer Böhme and Pascal Schöttle for the short but intense research collaboration we had in Innsbruck during the last phase of my PhD, hoping that the best is yet to come.

Finally, a deeply grateful thought goes to Francesco De Natale for the wise scientific guidance of the Multimedia Signal Processing and Understanding Lab, and his direct and indirect precious contributions to my PhD work.

Of course, the PhD life would have been much harder without the amazing people I always had around me. First, I really want to thank all the current and former MMLab members and friends, who shared with me these passionate years: Tien, Valentina, Paolo, Bo, Emanuele, Andrea, Nicola, Kashif, Tin, Habib, Krishna, Gwan, Alain, Mattia B, Alfredo, Mattia, thanks to all of you (and the ones that I might have forgotten) for the good company and the stimulating (research-related or not) discussions. Moreover, I want to reserve very special thanks to the fantastic people populating other DISI laboratories and offices, for the many coffee-break discussions and laughs which have been a valuable para-academic part of my daily work life. Then, I cannot forget the great colleagues that I met during my time abroad and always made me feel welcome: a thankful thought goes to the whole TSC-5 lab in Vigo and the Security and Privacy group in Innsbruck.

Finally, I want to thank all my friends and my wonderful family, who never failed to help and support me in these stormy years. I doubt I will ever be able to fully return what all of you gave me through the years. I can only say your unconditional love and support have been (and will always be) the foundations of all my choices and achievements.

Cecilia

Bibliography

[1] Reveal - Social Media Verification. http://revealproject.eu.

[2] Stock Historical Data - http://www.stockhistoricaldata.com.

[3] Ami corpus - http://groups.inf.ed.ac.uk/ami/corpus/, 2006.

[4] S. Akkoul, R. Lèdèe, R. Leconge, and R. Harba. A new adaptive switching median filter. *IEEE Signal Processing Letters*, 17(6), 2010.

[5] 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. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, 2011.

[6] I. Amerini, R. Becarelli, R. Caldelli, and M. Casini. A feature-based forensic procedure for splicing forgeries. *Mathematical problems in Engineering*, 2015.

[7] I. Amerini, R. Becarelli, R. Caldelli, and A.D. Mastio. Splicing forgeries localization through the use of first digit features. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 143–148, 2014.

[8] E. Ardizzone, A. Bruno, and G. Mazzola. Copy-move forgery detection by matching triangles of keypoints. *IEEE Transactions on Information Forensics and Security*, 10(10):2084–2094, 2015.

[9] K. Bahrami, A. C. Kot, L. Li, and H. Li. Blurred image splicing localization by exposing blur type inconsistency. *IEEE Transactions on Information Forensics and Security*, 10(5):999–1009, 2015.

[10] M. Barni and A. Costanzo. A fuzzy approach to deal with uncertainty in image forensics. *Signal Processing: Image Communuication*, 27:998–1010, 2012.

[11] M. Barni, M. Fontani, and B. Tondi. A universal technique to hide traces of histogram-based image manipulations. In *Proceedings of ACM Workshop on Multimedia and Security*, pages 97–104, 2012.

[12] M. Barni and F. Pérez-González. Coping with the enemy: advances in adversary-aware signal processing. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 8682–8686, 2013.

[13] M. Barni and B. Tondi. Multiple-observation hypothesis testing under adversarial conditions. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2013.

[14] M. Barni and B. Tondi. The source identification game: an information-theoretic perspective. *IEEE Transactions on Information Forensics and Security*, 8(3):450–463, 2013.

[15] M. Barni and B. Tondi. Binary hypothesis testing game with training data. *IEEE Transactions on Information Theory*, 60(8):4848–4866, 2014.

[16] J.R. Barry, E.A. Lee, and D.G. Messerschmitt. Digital communication. *Springer*, 2004.

[17] T. Bianchi and A. Piva. Analysis of non-aligned double JPEG artifacts for the localization of image forgeries. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 1–6, 2011.

[18] T. Bianchi and A. Piva. Detection of nonaligned double JPEG compression based on integer periodicity maps. *IEEE Transactions on Information Forensics and Security*, 7(2):842–848, 2012.

[19] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7, n. 3:1003–1017, 2012.

[20] T. Bianchi and A. Piva. Reverse engineering of double JPEG compression in the presence of image resizing. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 127–132, 2012.

[21] T. Bianchi, A. Piva, and F. Pérez-González. Near optimal detection of quantized signals and application to jpeg forensics. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2013.

[22] T.A. Birney and T.R. Fischer. On the modeling of DCT and subband image data for compression. *IEEE Transactions on Image Processing*, 4(2):186–193, 1995.

[23] R. Böhme, F.C. Freiling, T. Gloe, and M. Kirchner. Multimedia forensics is not computer forensics. In *ACM International Workshop on Computational Forensics (IWCF)*, pages 90–103, 2009.

[24] L.A. Breslow and D.W. Aha. Simplifying decision trees: A survey. *The Knowledge Engineering Review*, 12(1):1–40, 1997.

[25] G. Cao, Y. Zhao, R. Ni, and L. Yu. Forensic detection of median filtering in digital images. In *IEEE International Conference on Multimedia and Expo*, pages 89–94, 2010.

[26] H. Cao and A. C. Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4):899–910, 2009.

[27] Matthias Carnein, Pascal Schöttle, and Rainer Böhme. Forensics of high-quality JPEG images with color subsampling. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2015.

[28] T. Carvalho, H. Farid, and E. Kee. Exposing photo manipulation from user-guided 3-d lighting analysis. In *SPIE Symposium on Electronic Imaging*, 2015.

[29] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1), 2008.

[30] Y.L. Chen and C.T. Hsu. Image tampering detection by blocking periodicity analysis in JPEG compressed images. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 803–808, 2008.

[31] Y.L. Chen and C.T. Hsu. Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security*, 6, n. 2:396–406, 2011.

[32] P. Comesaña. Detection and information theoretic measures for quantifying the distinguishability between multimedia operator chains. In *Proc. of IEEE WIFS*, pages 211–216, 2012.

[33] P. Comesaña-Alfaro and F. Pérez-González. Optimal counterforensics for histogram-based forensics. In *Proceedings of IEEE ICASSP*, pages 3048–3052, 2013.

[34] P. Comesaña-Alfaro and F. Pérez-González. The optimal attack to histogram-based forensic detectors is simple(x). In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 1730–1735, 2014.

[35] V. Conotter, E. Bodnari, G. Boato, and H. Farid. Physiologically based detection of computer generated faces in video. In *IEEE International Conference on Image Processing (ICIP)*, pages 248–252, 2014.

[36] V. Conotter, P. Comesaña-Alfaro, and F. Pérez-González. Forensic detection of processing operator chains: recovering the history of filtered JPEG images. *IEEE Transactions on Information Forensics and Security*, 10(11):2257–2269, 2015.

[37] V. Conotter, D.T. Dang-Nguyen, M. Riegler, G. Boato, and M. Larson. A crowd-sourced data set of edited images online. In *ACM International Workshop on Crowdsourcing for Multimedia*, 2014.

[38] V. Conotter, H. Farid, and G. Boato. Detecting photo manipulation on signs and billboards. In *IEEE International Conference on Image Processing (ICIP)*, 2010.

[39] F.O. Costa, S. Lameri, P. Bestagini, Z. Dias, A. Rocha, M. Tagliasacchi, and S. Tubaro. Phylogeny reconstruction for misaligned and compressed video sequences. In *IEEE International Conference on Image Processing (ICIP)*, 2015.

[40] D. Cozzolino, G. Poggi, and L. Verdoliva. Efficient dense-field copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, 2015.

[41] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster: a new blind image splicing detector. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 1–6, 2015.

[42] D.T. Dang-Nguyen, G. Boato, and F.G.B. De Natale. 3D-model-based video analysis for computer generated faces identification. *IEEE Transactions on Information Forensics and Security*, 10(8), 2015.

[43] D.T. Dang-Nguyen, I.D. Gebru, V. Conotter, G. Boato, and F.G.B. De Natale. Counterforensics of median filtering. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 260–265, 2013.

[44] D.T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. RAISE - a raw images dataset for digital image forensics. In *ACM Multimedia Systems Conference (MMSys)*, pages 219–224, 2015.

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

[46] N. Khanna et al. Forensic techniques for classifying scanner, computer generated and digital camera images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1653–1656, 2008.

[47] W. Fan, K. Wang, F. Cayre, and Z. Xiong. A variational approach to JPEG anti-forensics. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3058–3062, 2013.

[48] Z. Fan and R. De Queiroz. Identification of bitmap compression history: JPEG detection and quantizer estimation. *IEEE Transactions on Image Processing*, 12, n. 2:230–235, 2003.

[49] H. Farid. Detecting digital forgeries using bispectral analysis. Technical report, MIT AI Memo, 1999.

[50] H. Farid. Exposing digital forgeries from JPEG ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009.

[51] H. Farid. Image forgery detection. *IEEE Signal Processing Magazine*, 26(2), 2009.

[52] X. Feng, I.J. Cox, and G. Doerr. Normalized energy density-based forensic detection of resampled images. *IEEE Transacations on Multimedia*, 14(3):536–545, 2012.

[53] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Reverse engineering of double compressed images in the presence of contrast enhancement. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 141–146, 2013.

[54] M. Fontani, E. Aragones-Rue, C. Troncoso, and M. Barni. The watchful forensic analyst: multi-clue information fusion with background knowledge. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2013.

[55] M. Fontani and M. Barni. Hiding traces of median filtering in digital images. In *European Signal Processing Conference (EUSIPCO)*, pages 1239–1243, 2012.

[56] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni. A framework for decision fusion in image forensics based on Dempster-Shafer theory of evidence. *IEEE Transactions on Information Forensics and Security*, 8, n. 4:593–607, 2013.

[57] M. Fontani, A. Bonchi, A. Piva, and M. Barni. Countering anti-forensics by means of data fusion. In *SPIE Media Watermarking, Security and Forensics*, 2014.

[58] J. Fridrich, M. Goljan, and R. Dui. Steganalysis based on JPEG compatibility. In *SPIE Multimedia Systems and Applications*, pages 275–280, 2001.

[59] D. Fu, Y.Q. Shi, and W. Su. A generalized Benford's law for JPEG coefficients and its applications in image forensics. In *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, 2007.

[60] L. Gaborini, P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro. Multi-clue image tampering localization. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 125–130, 2014.

[61] N.C. Gallagher and G.L. Wise. A theoretical analysis of the properties of median filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6), 1981.

[62] T. Gloe and R. Boehme. The Dresden image database for benchmarking digital image forensics. In *ACM Symposium on Applied Computing*, volume 2, pages 1585–1591, 2010.

[63] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme. Can we trust digital image forensics? In *ACM International Conference on Multimedia*, pages 78–86, 2007.

[64] S.D. Greenwald. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information.* PhD thesis, Harvard-MIT Division of Health Sciences and Technology, 1990.

[65] Anthony T.S. Ho. *Handbook of digital forensics of multimedia data and devices.* John Wiley & Sons, 2015.

[66] F. Huang, J. Huang, and Y.Q. Shi. Detecting double JPEG compression with the same quantization matrix. *IEEE Transactions on Information Forensics and Security*, 5(4):848–856, 2010.

[67] M.J. Huiskes and M.S. Lew. The MIR Flickr retrieval evaluation. In *ACM International Conference on Multimedia Information Retrieval*, pages 39–43, 2008.

[68] M. Iuliani, G. Fabbri, and A. Piva. Image splicing detection based on general perspective constraints. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 1–6, 2015.

[69] M.K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberrations. In *ACM Workshop on Multimedia & Security*, pages 48–55, 2006.

[70] J.M. Jolion. Images and benford's law. *Journal of Mathematical Imaging and Vision*, 14(1):73–81, 2001.

[71] X. Kang, M. Stamm adn A. Peng, and K.J. Ray Liu. Robust median filtering forensics using an autoregressive model. *IEEE Transactions on Information Forensics and Security*, 8(9):1456–1468, 2008.

[72] T. Kasparis and J. Lane. Adaptive scratch noise filtering. *IEEE Transactions on Consumer Electronics*, 39(4):917–922, 1993.

[73] E. Kee, J. O'Brien, and H. Farid. Exposing photo manipulation from shading and shadows. *ACM Transactions on Graphics*, 33(165):1–21, 2014.

[74] M. Kirchner and R. Böhme. Hiding traces of resampling in digital images. *IEEE Transactions on Information Forensics and Security*, 3(4):582–592, 2008.

[75] M. Kirchner and R. Böhme. Synthesis of color filter array pattern in digital images. In *SPIE Media Forensics and Security*, 2009.

[76] M. Kirchner and R. Böhme. Counter-forensics: attacking image forensics. *H.T. Sencar and N.D. Memon* Digital Image Forensics, 2013.

[77] M. Kirchner and J. Fridrich. On detection of median filtering in images. In *Proceedings of SPIE*, volume 7541, pages 101–112, 2010.

[78] H. Kobayashi, B.L. Mark, and W. Turin. *Probability, Random Processes and Statistical Analysis*. Cambridge University Press, 1991.

[79] Shi-Yue Lai and Rainer Böhme. Block convergence in repeated transform coding: JPEG-100 forensics, carbon dating, and tamper detection. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3028–3032, 2013.

[80] S.Y. Lai and R. Böhme. Countering counter-forensics: the case of JPEG compression. In *International Conference on Information Hiding*, 2011.

[81] B. Li, Y.Q. Shi, and J. Huang. Detecting doubly compressed JPEG images by using mode based first digit features. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 730–735, 2008.

[82] C.T. Li and Y. Li. Color-decoupled photo response non-uniformity for digital image forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, 22:260–271, 2012.

[83] J. Li, X. Li, B. Yang, and X. Sun. Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security*, 10(3):507–518, 2015.

[84] Q. Liu. Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery. In *ACM Workshop on Multimedia Forensics and Intelligence*, pages 25–30, 2011.

[85] Y. Liu, C. Liu, and D. Wang. A 1D time-varying median filter for seismic random, spike-like noise elimination. *Geophysics*, 74(1):17–24, 2009.

[86] J. Lukás and J. Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensics Research Conference*, 2003.

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

[88] W. Luo, J. Huang, and G. Qiu. JPEG error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3):480–491, 2010.

[89] W. Luo, Z. Qu, and G. Qiu. A novel method for detecting cropped and decompressed image block. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[90] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An overview on video forensics. *APSIPA Transactions on Signal and Information Processing*, 1, 2012.

[91] S. Milani, M. Tagliasacchi, and S. Tubaro. Discriminating multiple JPEG compression using first digit features. In *Proceedings of IEEE ICASSP*, pages 25–30, 2012.

[92] S. Milani, M. Tagliasacchi, and S. Tubaro. Antiforensics attack to Benford's law for the detection of double compressed images. In *Proceedings of IEEE ICASSP*, pages 3053–3057, 2013.

[93] G.B. Moody and R.G. Mark. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[94] P.M. Narendra. A separable median filter for image noise smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):20–29, 1981.

[95] R. Neelamani, R. De Queiroz, Z. Fan, S. Dash, and R.G. Baraniuk. JPEG compression history estimation for color images. *IEEE Transactions on Image Processing*, 15, n. 2:1363–1378, 2006.

[96] T. Ng and S.F. Chang. A model for image splicing. In *IEEE International Conference on Image Processing (ICIP)*, pages 1169–1172, 2004.

[97] T.T. Ng and S.F. Chang. Discrimination of computer synthesized or recaptured images from real images. *Digital Image Forensics*, Digital Image Forensics:1–36, 2012.

[98] A. Oliveira, P. Ferrara, A. De Rosa, A. Piva, M. Barni, S. Goldenstein, Z. Dias, and A. Rocha. Multiple parenting phylogeny relationships in digital images. *IEEE Transactions on Information Forensics and Security*, 11(2):328–343, 2016.

[99] T. Pander. The class of M-filters in the application of ECG signal processing. *Biocybernetics and Biomedical Engineering*, 26(4):3–13, 2006.

[100] C. Pasquini and G. Boato. JPEG compression anti-forensics based on first significant digit distribution. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 500–505, 2013.

[101] C. Pasquini, G. Boato, N. Anjalic, and F.G.B. De Natale. A deterministic approach to detect median filtering in 1D data. *IEEE Transactions on Information Forensics and Security*, 2016.

[102] C. Pasquini, G. Boato, and F. Pérez-González. Multiple JPEG compression detection by means of Benford-Fourier coefficients. In *IEEE Workshop on Informations Forensics and Security (WIFS)*, pages 113–118, 2014.

[103] C. Pasquini, G. Boato, and F. Pérez-González. Statistical detection of JPEG traces in digital images in uncompressed formats. *IEEE Transactions on Information Forensics and Security*, 2016, submitted.

[104] C. Pasquini, C. Brunetta, A.F. Vinci, V. Conotter, and G. Boato. Towards the verification of image integrity in online news. In *IEEE International Conference on Multimedia and Expo Workshop (ICMEW)*, 2015.

[105] C. Pasquini, P. Comesaña-Alfaro, F. Pérez-González, and G. Boato. Transportation-theoretic image counterforensics to First Significant Digit histogram forensics. In *Proceedings of ICASSP*, pages 2718–2722, 2014.

[106] C. Pasquini, F. Pérez-González, and G. Boato. A Benford-Fourier JPEG compression detector. In *IEEE International Conference on Image Processing*, pages 5322–5326, 2014.

[107] C. Pasquini, P. Schöttle, R. Böhme, G. Boato, and F. Pèrez-Gonzàlez. Forensics of high quality and nearly identical jpeg image recompression. In *ACM Information Hiding & Multimedia Security*, 2016, to appear.

[108] F. Pérez-González, G.L. Heileman, and C.T. Abdallah. Benford's law in image processing. In *IEEE International Conference on Image Processing*, pages 405–408, 2007.

[109] F. Pérez-González, T.T. Quach, S. J. Miller, C.T. Abdallah, and G.L. Heileman. Application of Benford's law to images. *S. J. Miller, A. Berger and T. Hill (Eds), The Theory and Applications of Benford's law, Princeton University Press*, 2014.

[110] T. Pevnỳ and J. Fridrich. Detection of double-compression for applications in steganography. *IEEE Transactions on Information Forensics and Security*, 3(2):247–258, 2008.

[111] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013.

[112] A.C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Information Forensics and Security*, 53(2):758–767, 2005.

[113] L.R. Rabiner, M.R. Sambur, and C.E. Schmidt. Application of a nonlinear smoothing algorithm to speech processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(6):552–557, 1975.

[114] S.T. Rachev and L. Ruschendorf. *Mass transportation problems*. Springer, 1998.

[115] S.O. Rice. Mathematical analysis of random noise. *Bell System Technical Journal*, 24:46–156, 1945.

[116] B. Rivet, L. Girin, and C. Jutten. Log-rayleigh distribution: a simple and efficient statistical representation of log-spectral coefficients. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3), 2007.

[117] A. Rocha, W. Scheirer, T.Boult, and S. Goldenstein. Vision of the unseen: current trends and challenges in digital image and video forensics. *ACM Computing Surveys*, 43(4), 2011.

[118] L. Rokach and O. Maimon. Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(4):476–487, 2005.

[119] A. De Rosa, A. Piva, M. Fontani, and M. Iuliani. Investigating multimedia contents. In *IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 1–6, 2014.

[120] G. Schaefer and M. Stich. UCID - An uncompressed colour image database. In *SPIE Storage and Retrieval Methods and Applications to Multimedia*, volume 5307, 2004.

[121] H.T. Sencar and N. Memon. Overview of state-of-the-art- in digital image forensics. In World Scientific Press, editor, *Statistical Science and Interdisciplinary Research*, 2008.

[122] H.T. Sencar and N. Memon, editors. *Digital Image Forensics - There is more to a picture than meets the eye*. Springer, 2013.

[123] M.C. Stamm, W.S. Lin, and K.J.R. Liu. Forensics vs anti-forensics: A decision and game theoretic framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[124] M.C. Stamm and K. J. Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6, no. 3:1050–1065, 2011.

[125] M.C. Stamm, M. Wu, and K.J.R. Liu. Information forensics: an overview of the first decade. *IEEE Access*, pages 167–200, 2013.

[126] A. Taddei, G. Distante, M. Emdin, P. Pisani, G.B. Moody, C. Zeelenberg, and C. Marchesi. The European ST-T Database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography. *European Heart Journal*, (13):1164–1172, 1992.

[127] M. Tao, Y.C. Liang, and F. Zhang. Resource allocation for delay differentiated traffic in multiuser ODFM systems. *IEEE Transactions on Wireless Communications*, 7(6):2190–2201, 2008.

[128] J.W. Tukey. *Exploratory Data Analysis*. MA: Addison-Wesley, 1976.

[129] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[130] F. Uccheddu, A. De Rosa, A. Piva, and M. Barni. Detection of resampled images: performance analysis and practical challenges. In *Signal Processing Conference (EUSIPCO)n*, pages 1675–1679, 2010.

[131] G. Valenzise, M. Tagliasacchi, and S. Tubaro. The cost of JPEG compression anti-forensics. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[132] G. Valenzise, M. Tagliasacchi, and S. Tubaro. Revealing the traces of JPEG compression anti-forensics. *IEEE Transactions on Information Forensics and Security*, 8(2), 2013.

[133] D. Vàzquez-Padìn, P. Comesaña-Alfaro, and F. Pérez-González. An svd approach to forensic image resampling detection. In *Signal Processing Conference (EUSIPCO)*, pages 2067–2071, 2015.

[134] David Vàzquez-Padìn, Pedro Comesana, and Fernando Pèrez-Gonzàlez. Set-membership identification of resampled signals. In *IEEE Workshop on Information Forensics and Security*, pages 150–155, 2013.

[135] Y. Wen and B. Zeng. A simple nonlinear filter for economic time series analysis. *Economics Letters*, 64(2):151–160, 1999.

[136] S.S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.

[137] J. Yang, J. Xie, G. Zhu, S. Kwong, and Y.Q. Shi. An effective method for detecting double JPEG compression with the same quantization matrix. *IEEE Transactions on Information Forensics and Security*, 9(11), 2014.

[138] J. Yang, G. Zhu, and J. Huang. Detecting doubly compressed JPEG images by factor histogram. In *Proc. of APSIPA*, 2011.

[139] H. Yuan. Blind forensics of median filtering in digital images. *IEEE Transactions on Information Forensics and Security*, 6(4):1335–1345, 2011.

[140] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris. Detecting image splicing in the wild (web). In *IEEE International Conference on Multimedia and Expo Workshop (ICMEW)*, pages 1–6, 2015.

[141] Y. Zhang, S. Li, S. Wang, and Y.Q. Shi. Revealing the traces of median filtering using high-order local ternary patterns. *IEEE Signal Processing Letters*, 21(3):275–280, 2014.

[142] X. Zhao, S. Wang, S. Li, and J. Li. Passive image-splicing detection by a 2-d noncausal Markov model. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(2):185–199, 2015.

[143] P. Zontone, M. Carli, G. Boato, and F.G.B. De Natale. Impact of contrast modification on human feeling: an objective and subjective assessment. In *IEEE International Conference on Image Processing (ICIP)*, 2010.

Appendix A

Derivation of $\sigma^2_{W_{r,i}}$

In order to obtain the variance of $W_{r,i}$, we need to consider the pdf of Z_q and study the real and imaginary parts of the r.v. $e^{-j\omega \log_{10} Z}$, i.e., the r.v.'s $C \doteq \cos(\omega \log_{10} Z_q)$ and $S \doteq -\sin(\omega \log_{10} Z_q)$, respectively. For the sake of simplicity, in the following analysis we drop the subscript q of Z. For deriving the pdf of C, the following r.v. transformations need to be applied

$$Z \xrightarrow{\log_{10}} Z' \xrightarrow{\cdot\omega} Z'' \xrightarrow{\cos} C$$

and the same happens for S by applying $-\sin(\cdot)$ as last transformation.

Since they are monotonic, the first two transformations can be treated with the formula

$$\begin{gathered} f_Y(y) = f_X(h^{-1}(y)) \cdot \left| \frac{\partial}{\partial y} h^{-1}(y) \right| \\ Y = h(X), \quad X \sim f_X(x) \end{gathered} \tag{A.1}$$

and we obtain

$$f_{Z''}(z'') = \mathcal{L}(10^{\frac{z''}{\omega}}) \cdot 10^{\frac{z''}{\omega}} \cdot \frac{\ln 10}{\omega}.$$

The third transformation is not monotonic and a generalization of formula (A.1) should be used [78]:

$$f_Y(y) = \sum_{\{x|h(x)=y\}} \frac{f_X(x)}{\left| \frac{\partial}{\partial x} h(x) \right|} \tag{A.2}$$

So, if we define $I_{Z''} = [\omega \log_{10}(q - q/2), \omega \log_{10}(q + q/2)[$, we have that

$$\begin{aligned} f_C(c) &= \sum_{\{z''|cos(z'')=c\} \cap I_{Z''}} \frac{f_{Z''}(z'')}{|-\sin z''|} \\ &= \sum_{D_c} \frac{e^{-\frac{\sqrt{2}}{\sigma}|10^{\frac{z''}{\omega}} - q|} 10^{\frac{z''}{\omega}} \ln 10}{\sigma\sqrt{2} N_{\sigma,q} \omega} \frac{1}{\sqrt{1-c^2}} \end{aligned}$$

where

$$\begin{aligned} D_c &= \{z''|\cos(z'') = c\} \cap I_{Z''} \\ &= \cup_{n\in\mathbb{Z}}\{2\pi n + t, 2\pi(n+1) - t\} \cap I_{Z''}, \\ &\qquad t \doteq \arccos(c). \end{aligned}$$

By definition, the variance of C is given by

$$\begin{aligned} \sigma^2_C &= \int_{-1}^{1} c^2 f_C(c)dc - \mu^2_C \\ &= \int_{-1}^{1} c^2 \sum_{z''\in D_c} \frac{e^{-\frac{\sqrt{2}}{\sigma}|10^{\frac{z''}{\omega}} - q|} 10^{\frac{z''}{\omega}}}{\sigma\sqrt{2}N_{\sigma,q}\omega} \frac{\ln 10}{\sqrt{1-c^2}} dc - \Re(a_{\omega,q})^2. \end{aligned}$$

Similarly, the variance of S is given by

$$\begin{aligned} \sigma^2_S &= \int_{-1}^{1} s^2 f_S(s)ds - \mu^2_S \\ &= \int_{-1}^{1} s^2 \sum_{z''\in D_s} \frac{e^{-\frac{\sqrt{2}}{\sigma}|10^{\frac{z''}{\omega}} - q|} 10^{\frac{z''}{\omega}}}{\sigma\sqrt{2}N_{\sigma,q}\omega} \frac{\ln 10}{\sqrt{1-s^2}} ds - \Im(a_{\omega,q})^2, \end{aligned}$$

where

$$\begin{aligned} D_s &= \{z''|\sin(z'') = c\} \cap I_{Z''} \\ &= \cup_{n\in\mathbb{Z}}\{2\pi n + t, 2\pi n + sign(t)\pi - t\} \cap I_{Z''}, \\ &\qquad t \doteq \arcsin(s). \end{aligned}$$

Finally, in order to obtain the variance of the real and imaginary parts of $W_{0,q} = \hat{A}_{\omega,q} - a_{\omega,q}$ we should consider that $W_{0,q}$ is a shift (i.e., has the same variance) of $\hat{A}_{\omega,q}$, which is the sample mean of $e^{-j\omega\log_{10} Z_q}$. By applying the Central Limit Theorem on both real and imaginary parts, we can obtain the two variances by dividing σ^2_C and σ^2_S by the number of samples M_q (see (3.16) and (3.17)).

Appendix B

Results for full decision trees

We report the results obtained in Sections 5.5.1 and 5.5.2 with the full decision trees for the two datasets considered.

B.0.1 Overall full decision tree

(a) Training set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.62	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.67	0.69	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.64	0.64	0.85	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.70	0.72	0.79	0.93	0.96	0.99	1.00	1.00	0.99	1.00	1.00
95	0.98	0.64	0.63	0.70	0.91	0.96	0.99	0.99	1.00	1.00	1.00	1.00
94	0.96	0.59	0.60	0.61	0.64	0.95	0.98	0.94	0.98	0.99	1.00	0.99
93	0.95	0.49	0.53	0.59	0.60	0.94	0.98	0.99	0.92	0.98	0.99	0.99
92	0.95	0.50	0.53	0.54	0.54	0.69	0.95	0.99	0.99	0.93	0.99	1.00
91	0.97	0.49	0.55	0.76	0.83	0.85	0.99	1.00	1.00	1.00	0.95	0.99
90	0.94	0.50	0.56	0.55	0.58	0.65	0.92	0.99	0.99	0.99	0.97	0.96

(b) Testing set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	0.91	0.95	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.70	0.44	0.63	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.54	0.45	0.46	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.54	0.49	0.47	0.89	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.45	0.61	0.60	0.75	0.96	0.67	1.00	1.00	1.00	1.00	1.00	1.00
95	0.35	0.56	0.58	0.66	0.93	0.94	0.77	0.98	0.99	1.00	1.00	1.00
94	0.36	0.48	0.48	0.51	0.49	0.98	0.98	0.57	0.98	1.00	1.00	1.00
93	0.39	0.36	0.32	0.46	0.47	0.97	0.99	0.98	0.45	0.99	0.99	1.00
92	0.32	0.34	0.40	0.37	0.40	0.64	0.97	0.99	0.98	0.58	0.99	1.00
91	0.32	0.34	0.42	0.72	0.84	0.89	1.00	1.00	1.00	0.96	0.56	0.99
90	0.33	0.36	0.39	0.43	0.52	0.58	0.94	1.00	1.00	0.99	0.96	0.53

Table B.1: Accuracies of overall full DT for UCID dataset.

(a) Training set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.68	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.99	0.68	0.66	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.68	0.63	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.74	0.78	0.75	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
95	0.96	0.71	0.67	0.80	0.79	0.98	1.00	1.00	1.00	1.00	1.00	1.00
94	0.99	0.81	0.78	0.81	0.86	0.99	1.00	0.99	1.00	1.00	1.00	1.00
93	1.00	0.71	0.75	0.79	0.85	0.94	1.00	1.00	0.94	1.00	1.00	1.00
92	0.94	0.66	0.71	0.71	0.70	0.64	0.96	1.00	1.00	1.00	1.00	1.00
91	1.00	0.58	0.59	0.69	0.73	0.73	0.93	1.00	1.00	1.00	0.99	1.00
90	0.99	0.76	0.77	0.82	0.74	0.75	0.96	0.99	1.00	1.00	1.00	0.98

(b) Testing set.

	NC	100	99	98	97	96	95	94	93	92	91	90
100	0.93	0.98	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.74	0.47	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.54	0.54	0.45	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.56	0.42	0.43	0.68	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.54	0.62	0.64	0.66	0.97	0.97	1.00	1.00	1.00	1.00	1.00	1.00
95	0.53	0.49	0.51	0.71	0.76	0.96	0.99	1.00	1.00	1.00	1.00	1.00
94	0.43	0.66	0.69	0.67	0.63	0.97	1.00	0.96	1.00	1.00	1.00	1.00
93	0.43	0.63	0.67	0.63	0.66	0.91	0.98	1.00	0.51	1.00	1.00	1.00
92	0.40	0.45	0.49	0.48	0.49	0.46	0.93	1.00	1.00	0.88	1.00	1.00
91	0.42	0.49	0.51	0.55	0.59	0.60	0.93	1.00	1.00	1.00	0.87	1.00
90	0.34	0.52	0.54	0.60	0.58	0.58	0.95	0.99	1.00	1.00	1.00	0.83

Table B.2: Accuracies of overall full DT for DRESDEN dataset.

B.0.2 QF$_c$-specific full decision trees

(a) Training set.

NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.61	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.65	0.66	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.63	0.67	0.86	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.78	0.69	0.82	0.94	1.00	0.99	1.00	1.00	1.00	1.00	1.00
95	1.00	0.73	0.73	0.81	0.95	0.98	0.99	1.00	1.00	1.00	1.00	1.00
94	1.00	0.77	0.80	0.74	0.79	0.98	1.00	0.99	0.99	0.99	1.00	1.00
93	0.98	0.66	0.73	0.74	0.73	0.98	1.00	1.00	0.98	0.98	0.99	1.00
92	0.97	0.48	0.61	0.60	0.58	0.75	0.96	0.99	0.98	0.97	0.99	1.00
91	0.95	0.47	0.53	0.76	0.79	0.88	0.99	1.00	1.00	0.99	0.94	1.00
90	0.95	0.52	0.51	0.68	0.73	0.71	0.96	1.00	1.00	1.00	0.99	0.95

(b) Testing set.

NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
99	0.77	0.43	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.58	0.43	0.47	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.52	0.49	0.54	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.41	0.61	0.63	0.74	0.95	1.00	1.00	1.00	1.00	1.00	1.00
95	0.43	0.62	0.64	0.71	0.95	0.98	1.00	0.99	1.00	1.00	1.00
94	0.38	0.64	0.63	0.68	0.64	0.99	0.99	1.00	0.99	1.00	1.00
93	0.42	0.57	0.58	0.65	0.67	0.98	1.00	0.99	0.97	0.99	0.99
92	0.40	0.38	0.44	0.45	0.45	0.69	0.97	1.00	0.98	0.55	0.99
91	0.38	0.32	0.33	0.73	0.80	0.89	1.00	1.00	1.00	0.95	0.56
90	0.30	0.44	0.51	0.50	0.66	0.66	0.96	1.00	1.00	1.00	0.99

Table B.3: Accuracies of QF_c-specific full DT for UCID dataset.

(a) Training set.

NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	1.00	0.61	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	1.00	0.51	0.54	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	1.00	0.57	0.65	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	1.00	0.71	0.76	0.77	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	1.00	0.78	0.74	0.82	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00
94	1.00	0.79	0.72	0.83	0.82	0.99	1.00	1.00	1.00	1.00	1.00	1.00
93	1.00	0.77	0.78	0.86	0.77	0.96	0.99	1.00	1.00	1.00	1.00	1.00
92	1.00	0.75	0.79	0.79	0.79	0.76	0.98	0.99	1.00	0.96	1.00	1.00
91	1.00	0.69	0.78	0.79	0.76	0.76	0.97	1.00	1.00	1.00	0.97	1.00
90	0.99	0.62	0.71	0.73	0.75	0.73	0.93	0.99	1.00	1.00	1.00	0.99

(b) Testing set.

NC	100	99	98	97	96	95	94	93	92	91	90
100	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
99	0.79	0.40	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
98	0.59	0.43	0.41	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
97	0.57	0.40	0.41	0.84	1.00	1.00	1.00	1.00	1.00	1.00	1.00
96	0.55	0.62	0.63	0.65	0.97	1.00	1.00	1.00	1.00	1.00	1.00
95	0.54	0.55	0.57	0.75	0.81	1.00	1.00	1.00	1.00	1.00	1.00
94	0.51	0.61	0.69	0.72	0.64	0.99	1.00	0.99	1.00	1.00	1.00
93	0.44	0.60	0.65	0.70	0.65	0.89	0.99	1.00	0.99	1.00	1.00
92	0.47	0.56	0.60	0.60	0.62	0.65	0.96	1.00	1.00	0.90	1.00
91	0.41	0.56	0.56	0.57	0.61	0.62	0.94	1.00	1.00	1.00	0.86
90	0.47	0.49	0.52	0.52	0.56	0.54	0.97	0.99	1.00	1.00	1.00	0.82

Table B.4: Accuracies of QF_c-specific full DT for DRESDEN dataset.