TECHNISCHE UNIVERSITÄT DARMSTADT
HOCHSCHULKENNZIFFER D-17
DEPARTMENT OF COMPUTER SCIENCE
SECURITY ENGINEERING GROUP

Security of Forensic Techniques for Digital Images

vom Fachbereich Informatik
der Technische Universität Darmstadt genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

MSc. Hieu Cuong Nguyen

geboren in Hai Duong, Vietnam

Erstreferent:	Prof. Dr. Stefan Katzenbeisser Technische Universität Darmstadt
Korreferentin:	Prof. Dr. Jana Dittmann Otto-von-Guericke-Universität Magdeburg

Tag der Einreichung: 25. Juni 2013
Tag der Prüfung: 15. August 2013

Darmstadt 2013

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, 25.06.2013

Hieu Cuong Nguyen

Abstract

Digital images are used everywhere in modern life and mostly replace traditional photographs. At the same time, due to the popularity of image editing tools, digital images can be altered, often leaving no obvious evidence. Thus, evaluating image authenticity is indispensable. Image forensic techniques are used to detect forgeries in digital images in the absence of embedded watermarks or signatures. Nevertheless, some legitimate or illegitimate image post-processing operations can affect the quality of the forensic results. Therefore, the reliability of forensic techniques needs to be investigated. The reliability is understood in this case as the robustness against image post-processing operations or the security against deliberated attacks.

In this work, we first develop a general test framework, which is used to assess the effectiveness and security of image forensic techniques under common conditions. We design different evaluation metrics, image datasets, and several different image post-processing operations as a part of the framework.

Secondly, we build several image forensic tools based on selected algorithms for detecting copy-move forgeries, re-sampling artifacts, and manipulations in JPEG images. The effectiveness and robustness of the tools are evaluated by using the developed test framework.

Thirdly, for each selected technique, we develop several targeted attacks. The aim of targeted attacks against a forensic technique is to remove forensic evidence present in forged images. Subsequently, by using the test framework and the targeted attacks, we can thoroughly evaluate the security of the forensic technique. We show that image forensic techniques are often sensitive and can be defeated when their algorithms are publicly known. Finally, we develop new forensic techniques which achieve higher security in comparison with state-of-the-art forensic techniques.

Zusammenfassung

Digitale Bilder werden überall im modernen Leben verwendet und ersetzen meist traditionelle Fotografien. Dabei können digitale Bilder oft ohne offensichtlich Beweise mit Hilfe von Bildverarbeitungwerkzeugen verändert werden. Deshalb ist die Überprüfung der Authentizität von Bildern unverzichtbar. Bildforensische Techniken werden verwendet, um Bildfälschungen in Abwesenheit von eingebetteten digitalen Wasserzeichen oder Signaturen zu erkennen. Dennoch können einige legitime oder illegitime Anwendungen der Bildnachbearbeitung die Qualität der bildforensischen Ergebnisse beeinflussen. Daher muss die Zuverlässigkeit forensischer Techniken untersucht werden. Die Zuverlässigkeit wird in diesem Fall als die Robustheit dieser gegen Operationen der Bildnachbearbeitung oder als die Sicherheit gegen zielgerichtete Angriffe verstanden.

In dieser Arbeit, entwickeln wir zunächst ein allgemeines Testframework, welches verwendet wird, um die Robustheit und Sicherheit der forensischen Techniken unter gemeinsamen Bedingungen zu messen. Wir entwerfen Metriken zur Auswertung, Wahrnehmung, Bilddatenbank, und verschiedene Operationen der Bildnachbearbeitung als Teil des Rahmenprogramms.

Zweitens erstellen wir mehrere forensische Werkzeuge auf Basis von ausgewählten forensischen Algorithmen zur Erkennung von Copy-Move (kopieren und verschieben) Fälschungen, Re-sampling der Bilder, und Manipulationen in JPEG Bildern. Die Leistung und Robustheit der forensische Werkzeuge werden mit dem entwickelten Testframework ausgewertet.

Als Drittes, entwickeln wir für jede ausgewählte Technik mehrere zielgerichtete Angriffe. Das Ziel zielgerichteter Angriffe ist es, forensische Beweise in gefälschten Bildern zu entfernen. Anschließend, können wir mit Hilfe des Testframeworks und der zielgerichteten Angriffe die Sicherheit der forensischen Techniken sorgfältig prüfen. Wir zeigen, dass bildforensische Techniken oft anfällig sind und besiegt werden können wenn ihre Algorithmen öffentlich bekannt sind. Schließlich entwickeln wir neue forensische Verfahren, die im Vergleich mit modernsten forensischen Techniken eine höhere Sicherheit erreichen.

Acknowledgements

To the people who contributed directly to the completion of my thesis, foremost, I would like to express my deepest gratitude to my advisor Prof. Dr. Stefan Katzenbeisser for his valuable support throughout my research at the Security Engineering Group. He showed me the right directions and helped me understand the essential things in doing a PhD thesis. I would not have completed this study without his devoted and patient guidance.

I would like to thank Prof. Dr. Jana Dittmann for being my second advisor and taking time to read and comment my thesis.

I would like to thank all my colleagues at the Security Engineering Group, Sami Alsouri, Sebastian Biedermann, Wolfgang Böhmer, Martin Franz, Nikolaos Karvelas, Martin Mink, Sascha Müller, Andreas Peter, Marius Senftleben, Andre´ Schaller, Heike Schröder, Marion Steiner, and Erik Tews. I am very happy to have an opportunity to work with them. I am thankful to Andrea Püchner and Heike Meissner for their organizational support.

I would like to express my thanks to the cooperation between the University of Transport and Communications in Hanoi and Darmstadt University of Technology. Within the framework of this close cooperation, I had an opportunity to conduct this study. I would like to extend my appreciation to the Ministry of Education and Training of Vietnam (MOET) and the German Foreign Academic Exchange Service (DAAD) for supporting me a scholarship for the three years. I would like to thank the Center for Advanced Security Research Darmstadt (CASED) and the Horst Görtz Stiftung (HGS) for the financial aid for the last year of my research.

Last, not least, I dedicate my special thanks to my family for giving me constant motivation and encouragement over the years.

Contents

1 Introduction 11

1.1 Motivation 11

1.2 Research Questions and Contributions 12

1.3 Thesis Organization 13

2 Preliminaries 15

2.1 Digital Image Watermarking 15

2.1.1 Requirements 16

2.1.2 Applications 17

2.2 Digital Image Forensics 18

2.2.1 Image Source Identification 21

2.2.2 Identification of Synthetic Images 22

2.2.3 Forgery Detection for Digital Images 23

2.3 Digital Image Anti-Forensics 24

2.3.1 Effectiveness and Security of Forensic Techniques 25

2.3.2 Anti-Forensic Techniques 25

3 A Framework for Evaluation of Image Forgery Detection 27

3.1 Introduction 27

3.2 The Proposed Framework 28

3.2.1 Framework Infrastructure and Evaluation Process 28

3.2.2 Performance Evaluation Metrics 30

3.2.3 Perceptual Evaluation Metrics 31

3.2.4 Possible Attacks 34

3.2.5 Image Datasets 35

3.2.6 The Test Tool 36

3.3 Summary 37

4 Security of Copy-Move Forgery Detection Techniques 39

4.1 Introduction 39

4.2 Effectiveness Analysis 41

4.2.1 Implementation of Detection Techniques 41

4.2.2 Effectiveness Analysis of Detection Techniques 43

4.3 Robustness and Security Analysis 44

4.4 A New Technique for Copy-Move Forgery Detection 48

4.4.1 Radon Transformation and Phase Correlation 48
4.4.2 The Proposed Technique 50
4.4.3 Experimental Results 51
4.5 Summary 54
5 Security of Re-sampling Detection Techniques 55
5.1 Introduction 55
5.2 Effectiveness Analysis 56
5.2.1 Implementation of Detection Techniques 56
5.2.2 Effectiveness Analysis of Detection Techniques 57
5.3 Robustness and Security Analysis 60
5.3.1 Robustness Analysis 60
5.3.2 Attack Based on Multiple Re-sampling 63
5.3.3 Attack Based on Hybrid Median Filter 64
5.3.4 Combination Attack 65
5.4 A New Technique for Re-sampling Detection 66
5.4.1 Fast Re-sampling Detection 66
5.4.2 The Proposed Technique 66
5.4.3 Experimental Results 68
5.5 Summary 72
6 Forgery Detection in JPEG Images 73
6.1 Introduction 73
6.2 Double JPEG Compression 75
6.2.1 DCT-based Compression 75
6.2.2 Double Quantization 77
6.2.3 Detection of Double JPEG Compression 78
6.2.4 Experimental Results 80
6.3 Security Analysis 81
6.3.1 Security of Double JPEG Compression Detection 81
6.3.2 Cropped Double JPEG Compression Detection 82
6.3.3 Experimental Results 83
6.4 A New Technique for Resized Double JPEG Compression Detection 85
6.4.1 The Proposed Technique 85
6.4.2 Experimental Results 86
6.5 Summary 90
7 Conclusions and Future Work 91
Appendix 93
A1. Determining the Last Quality Factor of a JPEG Image 93

List of Figures 95
List of Tables 97
List of Abbreviations 99
Bibliography 101

1 Introduction

1.1 Motivation

Photographs do not always tell the truth. In fact, the first image[1] forgeries appeared a long time ago, probably several years after Joseph Nipce produced the first photograph in 1825. For example, an iconic portrait of the US President Abraham Lincoln taken around 1860 is actually a forged image: the head of President Lincoln is depicted on the body of another person (see Figure 1-1). However, in the early days of photography, it was not easy to create forged images because making forgeries at that time required specific physical and chemical equipment and skills.

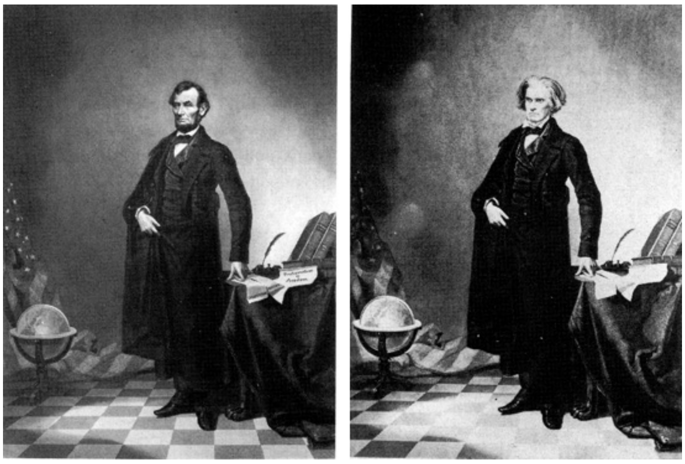


Figure 1-1: Shown on the left is the forged image of the US President Abraham Lincoln, which is a composite of the head of President Lincoln and a picture of the body of the southern politician John Calhoun (on the right)[2].

Nowadays, digital multimedia content (images, audio, video, etc.) can easily be created, stored, and transmitted. Digital images are ubiquitous in news, entertainment, science, financial documents, evidence in the court of law, etc. At the same time, since image editing tools are popular, making forgeries in digital images is an easy task. Even a novice can create a forged image without leaving obvious evidence that can be recognized by human eyes. Thus, the reliability of images became dubious and image authentication emerged as an important problem. There are many methods for digital image authentication, which can be divided into two main approaches, namely active and passive ones. The first approach consists of image watermarking methods and the second approach contains image forensic methods.

---

[1] In this thesis, the word *image* refers to natural photographs taken by cameras, unless otherwise mentioned.

[2] Most photographs in this thesis are courtesy of Hany Farid and the Darmouth Image Science Group.

Digital image watermarking is a popular method for image authentication, in which some additional information (called watermark) need to be embedded into an image during or after its creation. During detection, the watermark can be read and used for authentication. The major drawback of this approach is that watermarks need to be embedded in the image before distribution. However, most cameras in the market nowadays are not equipped with the function for watermark embedding (and this situation is unlikely to change in the near future) [1]. Thus, developing image authentication methods which do not rely on watermarks became an urgent need.

Image forensics is a passive method in which no information needs to be embedded prior to distribution. There are three main directions for image forensics research. The first direction identifies the sources of images, the second direction attempts to discriminate computer-generated images from natural images, and the third direction, probably the most important one, tackles the problem of forgery detection for digital images [1]. Since the problem of image forensics is very broad, our research focuses on forgery detection in digital images. Unfortunately, there is no universal technique that can detect every type of image forgery, thus many different image forensic techniques have been proposed, each of which comes with advantages and disadvantages. Therefore, the evaluation of forensic techniques for digital images has become an important problem.

1.2 Research Questions and Contributions

The main question addressed in this thesis is:

How can we evaluate the effectiveness and security of digital image forensic techniques?

To answer this question, several objectives have been achieved:

1. We developed a general test framework that allows a fair evaluation of image forensic techniques.
2. We built a number of image forensic techniques and applied the test framework to evaluate their effectiveness.
3. We designed targeted attacks against the selected image forensic techniques in order to assess their security and their resistance against attackers who aim at fooling the forensic tools.
4. We developed several new image forensic techniques for different types of image tampering, which overcome some limitations of existing algorithms.

1.3 Thesis Organization

The rest of the thesis is organized as follows:

In Chapter 2 we introduce important approaches for image authentication, both active and passive. In the active approach, we focus on image watermarking, its requirements and applications. In the passive approach, we provide a survey of main directions of image forensics, which aims at determining the source of images, distinguishing synthetic images from real images, and finding manipulations in images. In the last part of the chapter, we introduce the converse problem of forensics, namely anti-forensics. The goal of anti-forensics is to defeat forensic techniques. Nevertheless, anti-forensics can be used for assessment the security of forensic techniques. In this chapter, we define several concepts and terms, which we use in subsequent parts of the thesis.

In Chapter 3 we design a general framework for evaluation image forensic techniques. We define basic concepts and summarize the steps required to assess forensic techniques. We define several metrics that allow to measure performance of forensic techniques. Most metrics can be used for every technique, but some of them are suitable for a particular type of forensic technique. In addition, several general attacks are surveyed in this chapter.

In Chapter 4 we deal with copy-move forgery detection techniques. After studying existing techniques, we select three well-known ones for evaluation and improvement. Some targeted attacks are designed for each technique in order to assess the security. The evaluation results of the techniques are obtained by applying the framework of Chapter 3. Consequently, we design a new copy-move forgery detection technique. Chapter 4 is based on the papers:

- H.C. Nguyen, S. Katzenbeisser, "Security of Copy-Move Forgery Detection Techniques". In *36th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011).* IEEE Press, 2011.
- H.C. Nguyen, S. Katzenbeisser, "Detection of Copy-Move Forgery in Digital Images Using Radon Transformation and Phase Correlation". In *8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2012).* IEEE Press, 2012.

Chapter 5 deals with image re-sampling detection. When making forgeries, an image or a part of it is often resized or rotated. These geometric transforms involve a re-sampling step, which leaves detectable artifacts. Therefore, detecting traces of re-sampling became an important method to judge the authenticity of digital images. Similar to Chapter 4, we first survey state-of-the-art techniques for re-sampling detection and select three widely used techniques for evaluation within our test framework. Lastly, a new technique for re-sampling detection is proposed. Chapter 5 is based on the papers:

- H.C. Nguyen, S. Katzenbeisser, "Performance and Robustness Analysis for some Re-sampling Detection Techniques in Digital Images". In *10th International Workshop on Digital Forensic and Watermarking (IWDW 2011)*. Springer LNCS 7128, 2012.
- H.C. Nguyen, S. Katzenbeisser, "Robust Re-sampling Detection in Digital Images". In *13th International Conference on Communications and Multimedia Security (CMS 2012)*. Springer LNCS 7394, 2012.

Chapter 6 works with images compressed in JPEG format. When creating a forged JPEG image, it has to be loaded into a photo editing software, manipulated and then re-saved as JPEG. Therefore, detecting double JPEG compression is a significant step to authenticate JPEG images. However, when an image is cropped before recompression, detectable artifacts introduced by the JPEG compression algorithm will likely be destroyed. Thus, techniques for detection of cropped double JPEG compression are needed. In this chapter, we evaluate techniques for detecting double JPEG and cropped double JPEG compression. Lastly, we design a new technique to detect double JPEG compressed images even if they were resized before the second compression. Chapter 6 is based on the paper:

- H.C. Nguyen, S. Katzenbeisser, "Detecting Resized Double JPEG Compressed Images – Using Support Vector Machine". In *14th International Conference on Communications and Multimedia Security (CMS 2013)*. Springer LNCS 8099, 2013.

Chapter 7 concludes the thesis. The contributions of the thesis are highlighted, the benefits of this work are elaborated, and an outlook for future research is given.

2 Preliminaries

In this chapter, we briefly introduce image watermarking, image forensics, and anti-forensics. Watermarking is a method for digital image protection and authentication. This method requires to actively embed a watermark into images. Image forensics is a passive method, so it does not require any embedded information in images for authentication. The authenticity of an image can usually be determined based on the detection of alterations of intrinsic properties of images. Anti-forensics is a method to disguise illegitimate manipulations in images in order to deceive forensic techniques. Research on anti-forensics helps us to thoroughly understand the security of forensic techniques.

2.1 Digital Image Watermarking

Cryptography is the most commonly used method for protection digital data. Encrypted data can be protected and it is only accessible after decryption. Nevertheless, cryptography is not suitable for some applications such as multimedia data distribution, where data needs to be protected and traced even if it must be available in the clear [2].

Digital watermarking is a method to hide some watermarks into digital multimedia data (called cover data or host data), usually in a way that not easily be recognized by a human observer. The output of the watermarking scheme is watermarked data. After the embedding, the watermark can be detected or extracted from the watermarked data. The process of adding a watermark into the cover data is known as watermark embedding (Figure 2-1) and the converse process is known as watermark detection or watermark extraction (Figure 2-2). In order to strengthen the security, watermarking schemes usually use a secret key [2]. If the detection process needs the original image, it is called non-blind watermarking. Otherwise, it is called blind watermarking. Since blind watermarking is the most applicable, the term watermarking often implies to blind watermarking.

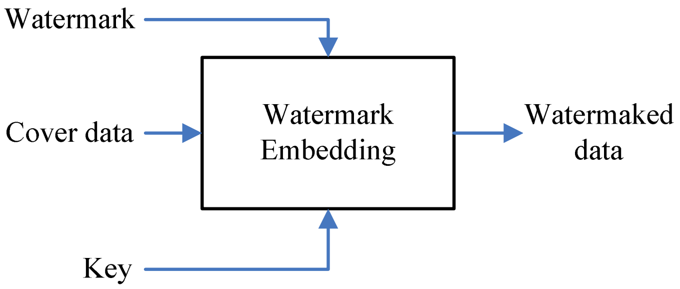


Figure 2-1: Watermark embedding process.

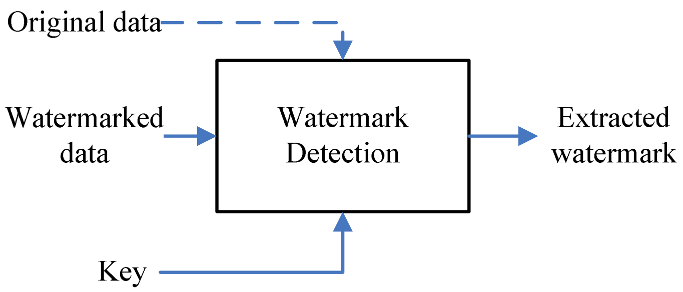


Figure 2-2: Watermark detection process.

Digital image watermarking has many applications (see Section 2.1.2), so different watermarking methods have been proposed. These methods are categorized into three main types: robust watermarking, fragile watermarking, and semi-fragile watermarking. They are briefly described as follows:

Robust watermarking: In this type, the most important requirement for watermarks is robustness against unintentional or malicious content manipulations. Robust watermarking is probably the most important type of watermarking and it is widely used in copyright management and digital content distribution.

Fragile watermarking: In contrast to robust watermarking, fragile watermarks have only limited robustness to a certain set of modifications. This type of watermarking is mostly applied to detect alterations of the watermarked image and is used for content authentication.

Semi-fragile watermarking: Since fragile watermarks are easily destroyed by any image manipulation, the incidental distortion by common post-processing operations can damage the watermark and render the image inauthentic. Semi-fragile watermarks are based on the image content instead of its digital representation. Thus, slight modifications caused by common image processing like mild JPEG compression, filtering, and contrast enhancement are accepted, meanwhile other manipulations which change the image content, like object addition, deletion and replacement can be revealed [3].

2.1.1 Requirements

There are some requirements that a watermarking system needs to satisfy. The importance of a requirement of a watermarking system always depends on the application. Nevertheless, some requirements are usually considered in watermarking systems: imperceptibility, robustness and security.

Imperceptibility: This is one of the most important requirements for watermarking systems: the watermarked content should be kept similar to the original content. In other words, the embedded watermarks should not create any unwanted artifact, causing visual quality degradation.

Robustness: This requirement means that the watermark cannot be removed or destroyed without visually degrading the content. The importance of this requirement is highly depends on the application.

Security: This is the ability to resist against deliberated attacks. It should be difficult for an adversary to remove or destroy a watermark without the knowledge of the secret key even if the watermarking algorithm is publicly known. For robust watermarking, any attempts to remove or destroy a watermark will severely degrade the visual quality of the image. For fragile watermarks, such attempts will destroy the authentication information [3].

2.1.2 Applications

Digital watermarking has many applications in different areas. Several applications of watermarking are listed in brief as follows [2]:

Copyright protection: This is probably the most important application of image watermarking. The embedded watermark can be recovered from an image and used to verify the authentication or ownership of the image. This application requires very high robustness: the watermark must not be destroyed, and in case more watermarks have been embedded, no ambiguity on the first inserted mark can be tolerated.

Copy protection: Digital content can carry watermarks consisting of copy information. This system is very useful for digital content distribution. For example, there were proposals to augment the DVD standard with copy information so that a disc can be read by a DVD player but no copies can be made.

Content authentication: The objective of this application is to detect modifications of digital content. For authentication applications, a so-called fragile watermark is embedded which will be destroyed once the content is modified. It should be noted that fragile watermarking requires the lowest level of robustness among all possible watermarking applications.

Fingerprinting: While in copyright protection, the same watermark is embedded into every copy; in fingerprinting, different copies carry watermarks. The watermarks in fingerprinting encode information on the legitimacy of a recipient of a copy instead of the source of the data.

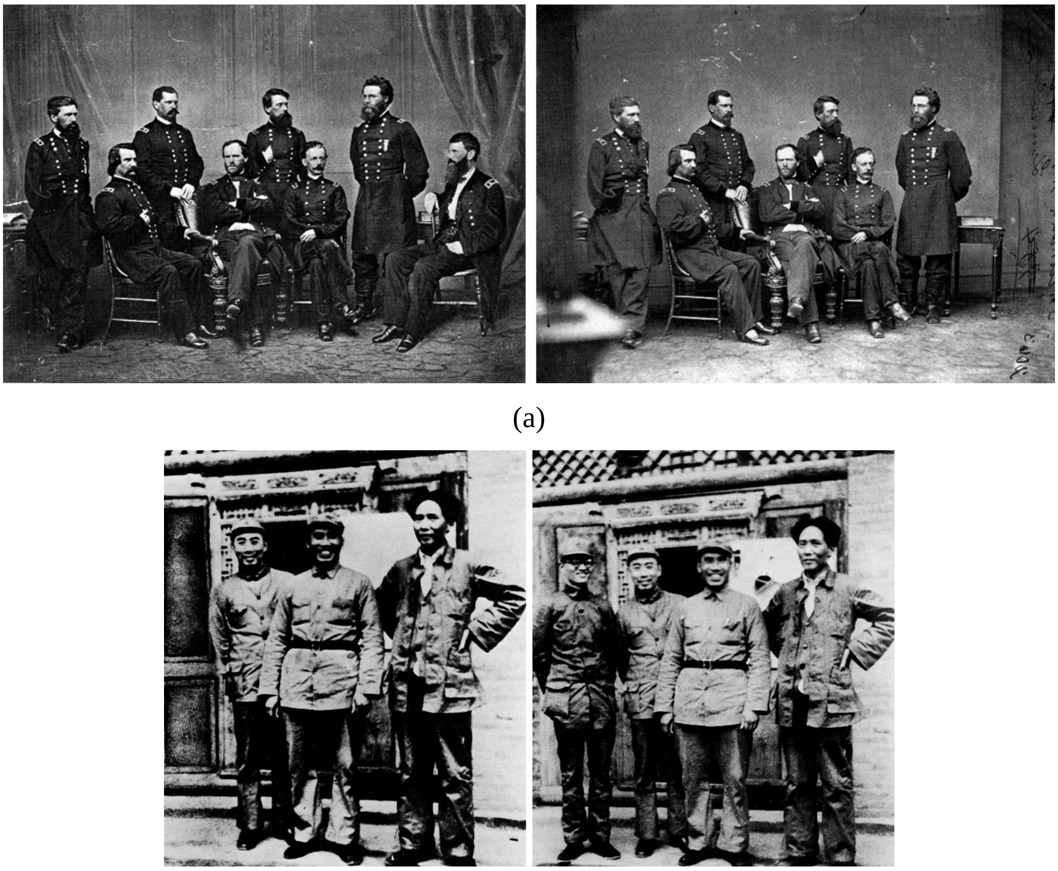
2.2 Digital Image Forensics

The drawback of digital watermarking is that it works only for the images where a watermark was embedded at the time of recording. Image forensics is a passive method, which can be used to detect image tampering without using an embedded watermark or any type of fingerprint. Image forensics is based on the assumption that although digital forgeries may leave no obvious evidence, they may alter intrinsic statistics of an image.

Image tampering has a long history and many examples of image tampering became known. For example, in a photograph made in circa 1865, General Francis P. Blair was added into the original photograph (Figure 2-3a). Due to the influence of photographs, they are often doctored because of political motives. Another example is shown in Figure 2-3b, where Po Ku had been removed from the left most position of the original photograph, after he fell out of favor with Mao Tse Tung [4].

Image tamperings do not only appear in politics, but also in many areas of everyday life. For example, image doctoring or retouching can be seen regularly on billboards, advertisements and magazine covers. There is no general rule when a modification must be considered an image tampering and it highly depends on applications. A slight doctoring or retouching is usually acceptable for most entertainment magazines. For example, the biceps of tennis player Andy Roddick were conspicuously enlarged on the cover of *Men's Fitness* magazine (Figure 2-4a). He commented that he was “pretty sure I’m not as fit as the *Men's Fitness* cover suggests”, but a spokesman for *Men's Fitness* said “We wouldn’t comment on any type of production issue. And I don’t see what the big issue is here” [4].

A little modification may be not a big issue in magazines such as *Man's Fitness* or *Star*, but it may be a more serious problem in scientific journals like *Nature* or *Science*. On the cover of *Nature* in August 2007 appeared three autonomous aircrafts taking atmospheric measurement. However, the top and the bottom aircrafts have been found to be copied and pasted (Figure 2-4b). After it was exposed, the editors had to print the following clarification: “The cover caption should have made it clear that this was a montage. Apologies” [4].

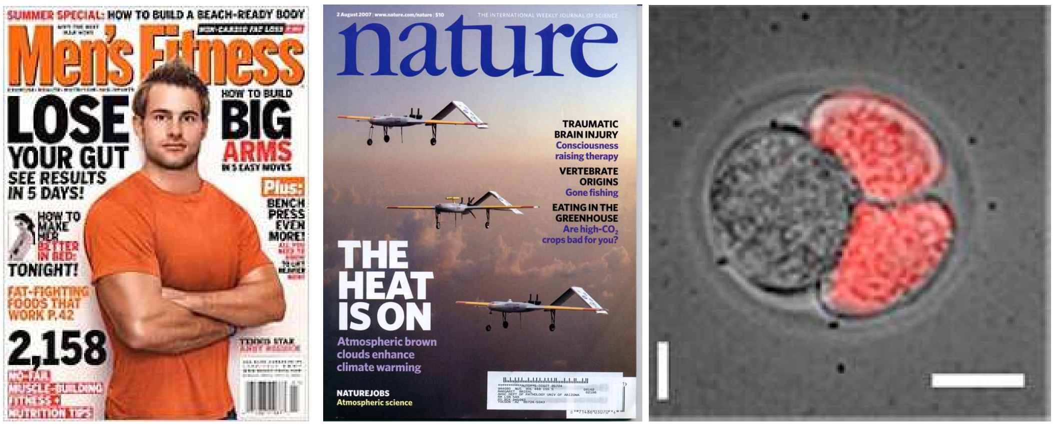


(a)

(b)

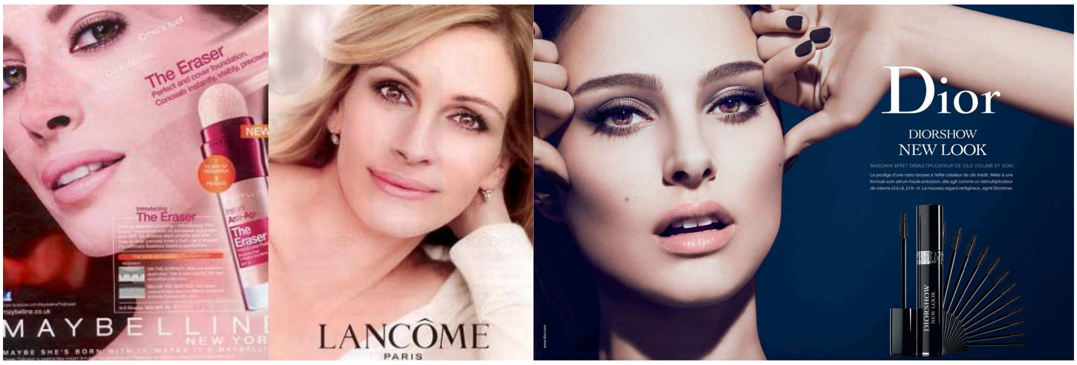
Figure 2-3: (a) Circa 1865: shown on the left is the forged photograph after General Francis P. Blair was added at the rightmost position and shown on right is the original photograph; (b) 1936: shown on the left is the forged photograph after removing Po Ku and shown on the right is the original photograph.

Image tampering is much more critical if it occurs in an image depicting scientific results. The Korean scientist Hwang Woo Suk has laid claim to a set of human-cloning patents, received a scientific excellence award, and published many papers. However, it became to know that in at least two of his papers published in the journal *Science* have been fabricated [5]. Missouri University professor Michael Roberts and co-authors published a paper (Cdx2 Gene Expression and Trophectoderm Lineage Specification in Mouse Embryos) in the journal *Science*. Contrary to conventional wisdom, the published research showed evidence that the first two cells of mouse embryos possess markers that indicate from a very early period whether they will grow into a fetus or placenta. However, an investigation uncovered that accompanying images (Figure 2-4c) of the paper were doctored. The authors had to withdraw the paper and explain their actions before a scientific committee [4].


SUMMER SPECIAL: HOW TO BUILD A BEACH-READY BODY
Men's Fitness
LOSE YOUR GUT
SEE RESULTS IN 5 DAYS!
HOW TO BUILD BIG ARMS
IN 5 EASY MOVES
Plus:
BENCH PRESS EVEN MORE!
HOW TO MAKE HER BETTER IN BED: TONIGHT!
FAT-FIGHTING FOODS THAT WORK P.42
2,158
NO-FAIL MUSCLE-BUILDING FITNESS + NUTRITION TIPS
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
TRAUMATIC BRAIN INJURY
Consciousness raising therapy
VERTEBRATE ORIGINS
Gone fishing
EATING IN THE GREENHOUSE
Are high-CO₂ crops bad for you?
THE HEAT IS ON
Atmospheric brown clouds enhance climate warming
NATUREJOBS
Atmospheric science

(a) (b) (c)

Figure 2-4: (a) 2007: the retouched photograph of Roddick in *Men's Fitness* Magazine; (b) 2007: a copy-move forged picture on the cover of *Nature*; (c) 2007: manipulated result of the group of professor Michael Roberts in a paper published in the journal of *Science*.



(a) (b) (c)

Figure 2-5: 2011: banned advertisements by the ASA of (a) Christy Turlington, (b) Julia Roberts and (c) Natalie Portmann.

Although not all areas require accuracy like in science, manipulations in images can affect their perception. In 2011, the British Advertising Standards Authority (ASA) banned two ads by cosmetics company L'Oreal due to excessive retouching. The first was an ad for Maybelline featuring Christy Turlington (Figure 2-5a) promoting a product called "The Eraser". The second was an ad for Lancome featuring Julia Roberts (Figure 2-5b), which claimed to "recreate the aura of perfect skin." In making their judgment on the Lancome ad, the ASA stated that they "could not conclude that the ad image accurately illustrated what effect the product could achieve, and that the image had not been exaggerated by digital post production techniques" [4]. In 2012 the ASA has banned another advertisement from appearing in any UK markets. The ad of Dior, featuring Natalie Portman (Figure 2-5c) was cited for being manipulated and Dior has agreed to withdraw this ad [6].

Along with forgery detection, other directions in image forensics, namely source identification and identification of synthetic images have been addressed in the scientific literature [1], [7]. In the following, we briefly review the main directions in image forensics.

2.2.1 Image Source Identification

The main goal of image source identification is to identify the digital image acquisition device (mostly the digital camera) or their characteristics (brand, model, etc.). Although some information on the acquisition device can be extracted from the image header file, this information can easily be modified or removed, therefore, it cannot reliably be used for the forensic purpose [1].

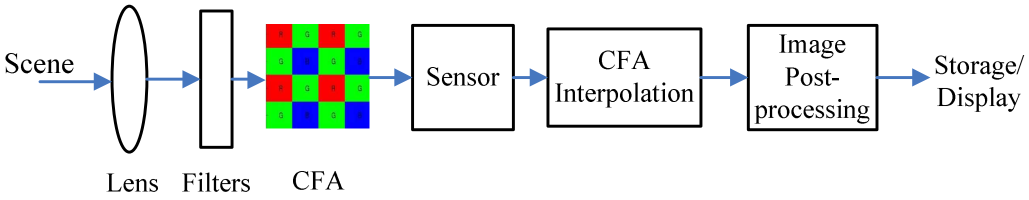


Figure 2-6: The pipeline for image capturing in digital cameras.

Digital cameras consist of a lens system, filters, Color Filter Array (CFA), imaging sensor, and a digital image processor (Figure 2-6) [8]. While taking a picture with a digital camera, the light reflecting the object runs through the lenses of the camera. After passing the lenses, the light goes through a set of filters, which are used to control the visible parts of the spectrum and reduce aliasing. The main part of a digital camera is the imaging sensor, consisting of an array of photodiode elements, or pixels, which convert light to analog signals. The signals are then converted to the digital domain and subsequently processed by the digital image processor. Digital cameras deploy Charge-Coupled Device (CCD) or Complimentary Metal-Oxide Semiconductor (CMOS) as imaging sensors. Sensor pixels are sensitive only to the brightness of light, thus producing a monochromatic output. To produce a color image, a CFA is used in front of the sensor so that each pixel records the light intensity of a single color. The output from the sensor is a mosaic of red, green, and blue pixels of different intensities. The measured color values are passed to the component of digital image processing, which applies several operations in order to produce a visual pleasant image [1], [9].

Each processing step may leave detectable artifacts, which are caused by specific characteristics of the corresponding components. Among them are the distortions of the lens, CFA interpolation, the imperfection of the imaging sensor, and artifacts of the color processing algorithms [1]. Some techniques extract specific features in images and then use them in classifiers in order to discriminate cameras. For example, the authors in [10] extract 34 features to fingerprint camera models and [11] uses 33 features of color, image quality, wavelet domain. Choi et al. [12] propose the lens radial distortion as a fingerprint to find the source camera. The choice of CFA and the details of the CFA interpolation are the most pronounced variations among different camera models [1]. Several techniques based on the features of CFA interpolation have been developed [13–15]. Geradts et al. [16] propose matching of CCD pixels and use them to determine the source camera. Sensor noise is mainly due to the imperfection of the imaging sensor resulting in slight difference between the captured scene and the image acquired by the camera [17]. The dominating component of sensor pattern noise is the Photo-Response Non-Uniformity (PRNU), because of the sensor manufacturing process, silicone inhomogeneities, and thermal noise [7]. In order to identify source camera, many techniques rely on PRNU have been developed [18–23]. An extension of sensor-based camera identification to images corrected for lens distortion is introduced in [24]. The PRNU noise features and CFA interpolation artifacts can be jointly used for detecting source type and camera model [25].

2.2.2 Identification of Synthetic Images

With the development of image processing and computer vision, computer-generated images can be created that are very similar to natural photographs. For example, a computer-generated image of a famous Korean actress (Song Hye Kyo), created by Indonesian artist Max Edwin Wahyudi, is shown in Figure 2-7. To create this image, he used a combination of digital sculpting, design application Pixelogic Zbrush and animation modelling software Autodesk 3DS Max [26]. The goal of this forensic direction is to discriminate natural photographs from synthetic images.

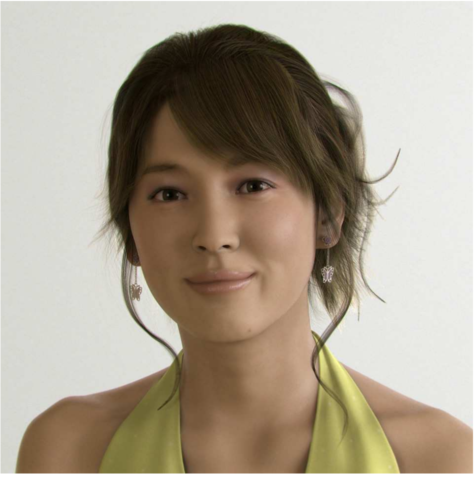


Figure 2-7: A computer generated picture of Song Hye Kyo.

The main idea of most of the existing work aiming at identifying synthetic images are to extract significant features from natural images and computer-generated images and use machine learning techniques for classification. Therefore, the most crucial difference between the techniques lies in the feature selection process. Lyu and Farid [27] use higher order statistics of wavelet transformation coefficients to design features. These features can be used to train a SVM-based classifier. Ng et al. [28] designed 192 features, which are based on identifying the distinctive characteristic of computer-generated images and natural images. The authors in [29] proposed a method, in which the features are obtained from characteristic functions of wavelet coefficients histograms. One can also exploit the fact that most real photographs are taken by digital cameras, which leave specific artifacts that do not exist in computer generated images. Thus, Dehnie et al. [30] proposed a method to discriminate synthetic images from digital camera images. Dirik et al. [31] extended the approach of [30] to include CFA interpolation artifacts.

2.2.3 Forgery Detection for Digital Images

Forgery detection is probably the most important problem of digital image forensics. Although many image manipulation methods have been proposed, there is no unique technique capable of detecting every forged image. We briefly revisit state of the art forgery detection techniques in some main categories, mostly following [32]:

Pixel-based methods: Since pixels are basic elements of digital images, analyzing pixel-level correlations can expose an image tampering. Copy-move (or cloning) is a common method for image tampering in order to conceal an object in the image. To detect this forgery, many techniques have been proposed. Most of existing copy-move forgery detection techniques relies on analyzing the specific features of image blocks, which are extracted by using different algorithms, such as DCT transform [18], DWT transform [33], FMT transform [34], and PCA analysis [35]. We will discuss this forgery type in more detail in Chapter 4.

Another common forgery is composition, where two or more images are spliced. The authors of [36], [37] show that splicing disrupts high-order Fourier statistics, which can be used to detect this forgery.

To create a convincing forged image, the image or its portions are usually resized or rotated. These geometric transforms require re-sampling, which consists of an interpolation step. Interpolation creates specific artifacts, which can be uncovered through analyzing the correlations of neighboring pixels [38], [39] or computing the second derivatives of the image [40], [41]. We will discuss this forgery type in more detail in Chapter 5.

Format-based methods: Most cameras encode images in the JPEG format. In order to create a forged image, the JPEG image is loaded into an imaging editor and it is re-saved as JPEG after modification have been performed. Thus, the forged JPEG image exhibits artifacts of double JPEG compression, which can be uncovered by some techniques [42][43]. It is noted that if the JPEG image has been cropped before the second compression, the corresponding JPEG blocking grids in the first compression and in the second compression are no longer aligned, so the aforementioned do not work. To detect this type of forgery, some more robust techniques have been proposed [44–46]. We will discuss techniques to detect forgeries in JPEG images in more detail in Chapter 6.

Camera-based methods: As presented in Section 2.2.1, digital cameras are equipped several components. Each of them may leave detectable artifacts, which are caused by specific characteristics of the corresponding components. The artifacts can be applied to determine camera source as well as image integrity. The CFA interpolation leaves forensic artifacts that can be used for detecting image manipulations [47–49]. Some other camera artifacts can be used as evidence of tampering, such as chromatic aberration [50] and sensor noise [51].

Physics-based methods: There are some techniques for estimating different properties in the lighting environment under which a person or an object was photographed. Thus, differences in lighting across the image can be used as tampering evidence [52], [53]. The lighting of a scene in practice can be complex due to different positions of the lights. A method to estimate a low-parameter representation of such complex lighting environments is described in [54]. Besides, inconsistencies in shadows can be used for tampering detection [55], [56].

Geometry-based methods: The principal point is the projection of the camera center onto the image plane, so it is moved proportionally when an object is translated in the image [32]. The authors of [57] use the inconsistencies in the principal point across an image as evidence of tampering. The discrepancies in motion blur in images have been used for detecting spliced images [58].

2.3 Digital Image Anti-Forensics

Anti-forensics is a method allowing to mislead forensic analysis of digital images. This method is usually used to assess the reliability of forensic methods, especially in the presence of an adversary that wants to influence the result of the forensic algorithm. Anti-forensics is also known as counter-forensics [59].

2.3.1 Effectiveness and Security of Forensic Techniques

While the *effectiveness* of a digital image forensic technique is the detection capacity of the technique in case no legitimate or illegitimate attack has been applied to forged images, the *robustness* of a digital image forensic technique is its reliability even if legitimate image post-processing is performed [59]. Most forensic techniques in the literature are tested with some common post-processing operations such as JPEG compression and Gaussian noise addition in order to measure their reliability. The authors of [60] show that the common manipulations allow to judge the reliability of the forensic techniques only on an average. In fact, based on the knowledge of a forensic technique, which are mostly published, adversaries can design deliberated attacks in order to defeat the technique.

The *security* of a digital image forensic technique is defined by its reliability to detect forgeries even in case intentionally concealed illegitimate post-processing has been applied to forged images [59]. In other words, security is the ability to withstand anti-forensics. Thus, the security of an image forensic technique can be evaluated by examining its resistance against targeted attacks.

In the next section, we briefly introduce several anti-forensic techniques, which are used to assess the security of different forensic techniques.

2.3.2 Anti-Forensic Techniques

At present, only a few anti-forensic techniques have been proposed. One of the earliest digital image anti-forensic techniques was introduced by [60]. The technique has successfully destroyed the traces of re-sampling, which are caused by image resizing or rotation. To hide fingerprints left by image re-sampling, a set of targeted attacks have also been proposed in [61]. Some other anti-forensic techniques try to forge the PRNU noise of camera sensor left in images [62] and to artificially synthesize CFA artifacts [63]. Stamm et al. [64], [65] proposed methods to remove quantization artifacts from the DCT coefficients of JPEG compressed images and from the wavelet coefficients of wavelet-based compression schemes such as Set Partitioning in Hierarchical Trees (SPIHT) and Embedded Zero-tree Wavelet (EZW).

While anti-forensics can defeat forensic techniques, some anti-forensic operations may leave detectable evidence of their own. Detecting traces of anti-forensic operations can uncover the presence deliberated attacks as well as help to improve forensic techniques. Since the median filter is used in some anti-forensic techniques [60], [61], detecting traces of median filtering can uncover the evidence of possible attacks [66]. The authors of [67] show that how such anti-forensic techniques [64], [65] affect the visual quality of JPEG images.

3 A Framework for Evaluation of Image Forgery Detection

3.1 Introduction

Many forgery detection techniques for digital images have been proposed in the literature. Therefore, there is a need to evaluate forgery detection techniques in a controlled environment in order to assess their performance. The purpose of evaluation is two-fold. Firstly, it provides either a qualitative or a quantitative method of evaluating a technique. Secondly, it allows to compare different techniques under similar criteria [68]. So far, most existing detection techniques were only tested independently, it is difficult to reproduce their experimental results and compare the techniques to each other. Therefore, in order to test detection techniques in an efficient and comparable way, we propose a general framework that uses common evaluation conditions of image datasets, evaluation metrics, and attacks.

As mentioned in Chapter 2, there are two main approaches for image authentication: the active approach using watermarking techniques [2] and the passive approach involving image forensic techniques [32]. For the evaluation of watermarking techniques, several standard frameworks or benchmarking systems have been proposed such as Stirmark [69], Checkmark [70], and Optimark [71], etc. In these systems, the watermarking technique under test is used to embed watermarks into several host images. The major requirement of embedded watermarks is to remain detectable even if the watermarked images have been altered. To measure this robustness, different manipulations are applied to watermarked images before they are fed into the detector. The output of watermark detector is used to evaluate the effectiveness and robustness of the analyzed technique. In order to obtain reliable results, one should perform multiple trials with different watermarks and images of various sizes and contents. When building benchmarking systems, the essential components are evaluation metrics and the set of image manipulations or possible attacks [72]. An important problem in the evaluation of watermarking techniques is to assess the perceptual quality of an image that has been watermarked or attacked. There is a tradeoff between the watermark embedding strength and the visual quality of the image. Since there is no universal metric for evaluation of perceptual quality, different metrics are usually considered in the benchmarking systems. The Peak Signal to Noise Ratio (PSNR) is one of the most popular metrics for perceptual quality evaluation.

In the field of image forensics, each forgery detection technique is usually assessed independently. Some authors have tried to compare detection techniques of the same type [73–77]. The authors of [73] evaluate three copy-move forgery detection techniques based on the Discrete Cosine Transform (DCT), Principal Component Analysis (PCA), and the Fourier-Mellin Transform (FMT). They test the robustness of the techniques against common image manipulations such as JPEG compression, rotation and scaling. The effectiveness and robustness of some re-sampling detection techniques have been analyzed in [74]. The effectiveness and robustness of several copy-move forgery detection techniques were measured in [75], [77] and some DCT-based forgery detection techniques were evaluated in [76]. All mentioned works used empirical methods to test a group of techniques in the same category under the same condition. However, they did not propose an evaluation framework that other people can subsequently use.

In this chapter, we describe a general framework for evaluating the effectiveness and security of image forgery detection techniques. To this end, we introduce the attack models and the infrastructure of the evaluation system. In addition, we design a test tool in order to support the evaluation in practice. With the framework, all analyzed techniques can be tested under the same condition, which therefore allows fair comparisons. The framework will be used for evaluating the selected forensic techniques; results are shown in the next chapters.

3.2 The Proposed Framework

3.2.1 Framework Infrastructure and Evaluation Process

The main goal of the proposed framework is to empirically evaluate the effectiveness and security of image forgery detection techniques under a common condition. In forgery detection techniques, the input is the to-be-tested image and the output is a decision indicating whether the image is forged. Subsequently, in the framework for evaluation of detection techniques, the input is the tested technique and the output is an evaluation report describing the effectiveness and security of the techniques. The common infrastructure and evaluation process of the proposed framework are shown in Figure 3-1. The infrastructure consists of several components of evaluation metrics, possible attacks, and image datasets. These components are necessary for the process of testing a detection technique and they will be briefly described in the next sections.

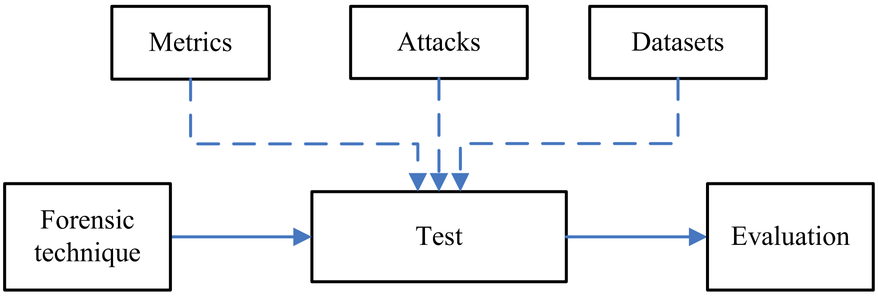


Figure 3-1: The infrastructure of the proposed framework.

As mentioned in Section 2.3.1, the security of a forensic technique can be evaluated by assessing the resistance of the technique against targeted attacks. There are two attack models for forensic techniques:

- **Blind attack model**: In this model, the adversary does not know the algorithm of the detection technique; he sees the forensic technique as a black box. Thus, the technique can only be attacked by using common image manipulations. With respect to evaluating the security of a forensic technique, these attacks are not enough to gain a reliable security evaluation.
- **Non-blind attack model**: In this model, the adversary knows the forensic algorithm in detail, so he can design targeted attacks against the technique. Since the adversary can utilize knowledge of the forensic technique, this allows tests under more stringent conditions. In fact, most existing image forensic algorithms are published in the literature; the non-blind attack model is thus more realistic and applicable.

The evaluation process of the framework is shown in more detail in Figure 3-2. To evaluate the effectiveness of a forensic technique, the detection processes are applied to a dataset of forged images. A forged image is created by making forgeries to an original image. Since there are many types of forgeries, many different datasets of forged images need to be created. The security of a technique is assessed by applying the detection processes to datasets of attacked images. An attacked image is created by applying targeted attacks to a forged image. An attack against a forensic technique is considered successful if the technique detects the attacked image as original.

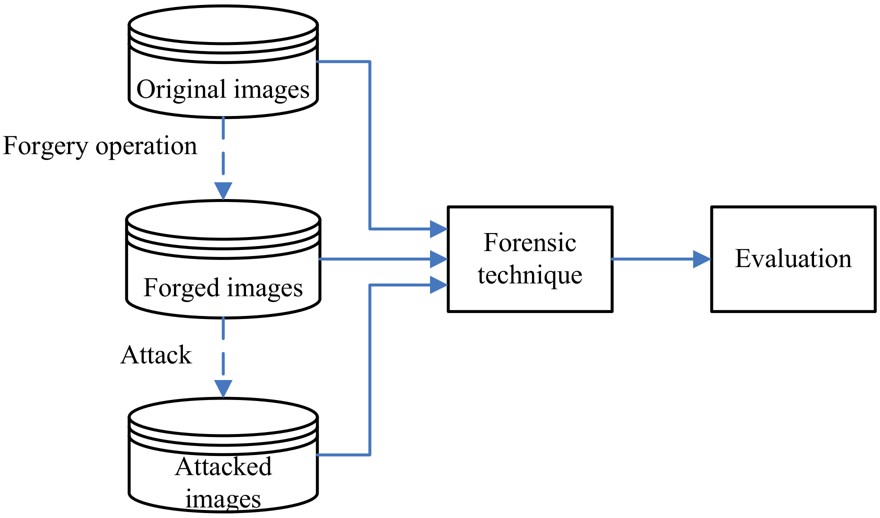


Figure 3-2: The evaluation process of the framework for assessment of forensic techniques.

3.2.2 Performance Evaluation Metrics

A performance metric is a meaningful and computable measure used for quantitatively evaluating the performance of any forgery detection technique [68]. In this section, we first revisit the common metrics of *true positive rate* and *false positive rate*. Then we define some metrics which are designed for a specific type of forgery, such as the *correctness rate* and the *incorrectness rate*.

The outcome of a forensic technique is binary: either positive (predicting that the image is forged) or negative (predicting that the image is original). The test results for each image may or may not match the actual status of the image. Thus, we can consider *true positives*, where a forged image is correctly identified as forged), *false positives*, where an original image is incorrectly identified as forged, *true negatives*, where an original image is correctly identified as original, and *false negatives*, where a forged image is incorrectly identified as original.

The true positive rate *(TPR)* and the false positive rate *(FPR)* are defined as

$$TPR = \frac{TP}{TP + FN} \ ,$$

$$FPR = \frac{FP}{TN + FP} ,$$

where TP, TN, FP, FN are the number of true positives, true negatives, false positives and false negatives respectively.

The *detection rate* is the fraction of the number of images detected as forged and the total number of testing images. In a test with a dataset of all forged images, the true positive rate is equal to the detection rate. Similarly, in the test with a dataset containing only original images, the *false positive rate* is computed as the fraction of the number of original images which have been detected as forged and the total number of testing images.

While the detection rate and false positive rate are general metrics, some other evaluation metrics are only suitable for a certain forgery type. Copy-move forgery is a very popular problem in image forensics, where an image is judged as forged if there are two similar regions in the image. Typically, these algorithms are able to identify the copied regions pretty accurately. However, the detection algorithm may produce false positives when the detected results are, for example, parts of homogeneous image regions or produce errors when estimating the forged regions. In order to evaluate the accuracy of detection techniques, we use the so-called *correctness rate*, which described as follows.

Assuming that D_1 and D_2 are the copied parts in the tested image; R_1 and R_2 are the two similar image regions which were detected by the forensic technique. The accuracy of the technique based on D_1, D_2, R_1, and R_2 is evaluated by computing a metric C as follows:

$$C = \frac{|R_1 \cap D_1| + |R_2 \cap D_2|}{|D_1| + |D_2|} \; .$$

Assuming that the number of testing images is N and the number of images correctly detected as forged is Nc, the *correctness rate (CR)* is defined as:

$$CR = \frac{Nc}{N} \; .$$

Consequently, the *incorrectness rate (ICR)* is defined as:

$$ICR = \frac{Nc - Nf}{N} \; ,$$

where Nf denotes the number of images detected as forged.

3.2.3 Perceptual Evaluation Metrics

To assess the security of a detection technique, different attacks against the technique must be used. At the same time, the attacks usually degrade the perceptual quality of the attacked images. A good attack not only deceives the detection technique, but it creates as little impact to image visual quality as possible. For example, it is unacceptable if an attack deceives a forensic technique but also distorts images so much that the attack can be easily recognized by human eyes. There is usually a tradeoff between the strength of an attack and the perceptual quality of attacked images. For a fair benchmarking of image forensics, the perceptual quality loss due to an attack is an important issue that should be considered. Although there are many metrics for the evaluation of image visual quality, none of them is universal. Therefore, in this section, we review common metrics which take the effects of the Human Visual System (HVS) into account; some of them will be used for perceptual quality evaluation in next chapters.

One of the most popular perceptual quality metrics for digital images is the Mean Square Error (MSE). The MSE is the mean of the squared error values across the entire image between an image I and its manipulated version K (of the same size of $M\times N$) and it can be defined as follows:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j)-K(i,j)\right)^2 .$$

The Signal to Noise Ratio (SNR) is defined as the power ratio between a signal (meaningful information) and noise (unwanted signal). This metric is useful to quantify how much noise is contained in an image. The larger the SNR is, the better the quality of the manipulated image. The SNR can be computed as follows:

$$SNR = 10\log_{10}\left(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j)-K(i,j)\right)^2}\right)$$

$$= 10\log_{10}\left(\frac{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)^2}{MSE}\right) .$$

A more popular (and widely-used) version of the SNR is the Peak Signal to Noise Ratio (PSNR). The PSNR is the ratio between the maximum possible power of a signal (or the peak value of the input image, called MAX_I) and the power of corrupting noise that affects the fidelity of its representation. In gray-scale images, when the pixels are represented using 8 bits per sample, $MAX_I = 255$. The PSNR can be calculated based on the mean square error as follows:

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) .$$

The SNR and PSNR are usually measured in decibel (dB). Although these metrics are very popular and simple to calculate, they are not always correlated to human vision [78]. Thus, better methods for image perceptual quality evaluation have been proposed. Wang et al. [79]

proposed an improved approach called Structural Similarity Index Metric (SSIM). It is based on the fact that the HVS is highly adapted for extracting structural information. Another approach is the Weighted Peak Signal to Noise Ratio (WPSNR) first introduced in [80]. Based on the fact that the human eyes are less sensitive to modifications in textured areas than in smooth areas, the WPSNR uses an additional parameter called the Noise Visibility Function (NVF), which is a texture masking function. The WPSNR of an image can be calculated as follows [80]:

$$WPSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE * NVF^2}\right).$$

The NVF uses a Gaussian model to estimate how much texture exists in any area of an image. For flat regions, the NVF is close to 1; for edges or textured regions, the NVF is close to 0. Thus, for smooth images, WPSNR approximately equals to PSNR, but for highly textured image, WPSNR is higher than PSNR. The function NVF at a pixel (i, j) is given as:

$$NVF(i,j) = \frac{1}{1+\theta\sigma^2(i,j)} ,$$

where $\sigma^2(i,j)$ denotes the local variance in a window of size $(2L+1)\times(2L+1)$ centered around the pixel with coordinate (i, j) and θ is a tuning parameter dependent on the image. The local variance is computed as:

$$\sigma^2(i,j) = \frac{1}{(2L+1)^2}\sum_{k=-L}^{L}\sum_{l=-L}^{L}(x(i+k,j+l)-\bar{x}(i,j))^2 ,$$

with

$$\bar{x}(i,j) = \frac{1}{(2L+1)^2}\sum_{k=-L}^{L}\sum_{l=-L}^{L}x(i+k,j+l).$$

The tuning parameter is given as:

$$\theta = \frac{D}{\sigma_{max}^2} ,$$

where σ_{max}^2 is the maximum local variance for a given image and D is an experimentally determined parameter that ranges from 50 to 100.

The value of NVF for an image of size $M\times N$ can be computed as the normalization of the noise visibility function of every image pixel:

$$NVF = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} NVF(i,j)^{2}} \quad .$$

In our framework, we use both metrics PSNR and WPSNR for perceptual quality evaluation (the first metric due to its popularity and the second metric due to its better correlation to human vision).

3.2.4 Possible Attacks

As mentioned in Section 3.2.1, attacks can be classified as blind and non-blind. In the former case, the detection algorithm is not known to the attacker, so the technique can only be attacked by general image manipulations. In the latter case, targeted attacks can be designed to defeat a certain detection technique. In other words, within a benchmark tool we can consider general attacks and targeted attacks.

General attacks consist of common image processing operations which may destroy forensic evidence and are applicable to any forensic tool. Some popular operations such as geometric transformations, JPEG compression, and Gaussian noise addition are usually considered in this category. Targeted attacks are specifically tailored towards a particular detection technique. Since targeted attacks are closely related to the technique which they affect, we will discuss them in more detail in the following chapters. In this section, we list important image manipulations which are implemented as general attacks in our framework. Some of them have already been used in the watermarking benchmarking systems of Stirmark [69] and Checkmark [70]. Various common attacks are described in brief as bellows:

A. Geometric transformations

- **Rotation**: Rotating the whole image or a part of it with a small angle. This operation is not easily recognized for a human observer, but it can affect the position of forensic evidence in the image.
- **Rotation and cropping**: Rotating the whole image and then cropping out a rectangular part from the rotated image which belongs to the original image.
- **Rotation, cropping and rescaling**: Rotating the whole image, cropping out a rectangular part from the rotated image which belongs to the original, and then rescaling the cropped part to the same size of the original image.
- **Scaling**: This attack can be divided into two groups: non-uniform scaling and uniform scaling. Non-uniform scaling uses different factors in horizontal and vertical directions.

Under uniform scaling, the scaling factor in horizontal and vertical are identical. In experiments, one often uses uniform scaling.

- **Cropping**: Cropping a small number of pixels from the edges of images. This manipulation is not easily visually recognized, but it can be an effective attack, especially against techniques which work by detecting image blocking artifacts.

B. Image Enhancing

- **Mean filtering**: is a simple filtering method used for image smoothing. It is often used to reduce noise in images.
- **Median filtering**: is a popular filtering method to reduce noise in images. It is often better than the mean filter since it preserves useful details in images.
- **Histogram modification**: is a method for adjusting image intensities to enhance contrast of images.
- **Gamma correction**: This is used to enhance images or adapt images for display.

C. Noise addition: Typically, additive white Gaussian noise is added to the whole image (globally) or to a part (locally).

D. JPEG compression: This is very popular lossy compression, which reduces invisible details in images. It is usually considered as an important attack against many forensic techniques. The degree of compression can be adjusted and there is a tradeoff between the compression factor and image quality.

To make convincing forgeries, several different attacks can be combined. For example, cropping can be combined with other operations such as rescaling to retain the size of the image. Rotation and scaling alone are sometimes not enough to defeat a detector and are used in combination with JPEG compression [81].

3.2.5 Image Datasets

To empirically evaluate the effectiveness and security of a forensic technique, the technique is tested on different types of images. In this section, we introduce a method to create the necessary datasets of forged images and attacked images. Firstly, a dataset of uncompressed original images forms the basis of our test; we choose a dataset of original images from the Uncompressed Color Image Database (UCID) [82]. This dataset consists of 1338 uncompressed images, including photos of natural scenes and objects, both indoor and outdoor. Besides, the UCID dataset is widely-used, free and can easily be downloaded from the Internet. A database consisting of original single JPEG compressed image is created by compressed the uncompressed original dataset with different quality factors.

Following the proposed evaluation process, in order to evaluate the effectiveness of a technique, we have to run tests on datasets of forged images. Since there are many forgery types, we prepared different forged datasets respectively. Forged datasets are created by applying some forgery manipulations to the dataset of original images and some of them are listed as follows:

- **Copy-move**: copy a random part of the original image and move it to another non-overlapping position in the same image. Since the size and location of the copied parts can affect the detection result, we use both squared and non-squared regions of various sizes when creating forgeries.
- **Re-sampling**: apply geometric transformations (e.g. up-sampling, down-sampling, rotation etc.) with different factors to original images in order to create re-sampled images.
- **Double JPEG compression**: double JPEG compressed images are created by applying JPEG compression with different quality factors one more time to a single JPEG compressed image.

3.2.6 The Test Tool

In this section, we briefly introduce a test tool based on the proposed framework in order to assess forgery detection techniques. The main purpose of this tool is to support the evaluation of forensic techniques in practice. To this end, we developed a set of Matlab functions, which are divided into three main groups: 1) attack functions, 2) functions for creating datasets of forged images and attacked images and 3) test functions.

The tool stimulates the detection process of a forensic technique on different image datasets. It applies a series of tests to different image datasets of forged images and attacked images. Each test is accomplished by applying the detection function to an image. The names of the image datasets to be used are parameterized in a profile, specified by users. Subsequently, the detection results of the forensic technique on the analyzed datasets are obtained from a report file. In order to accomplish an evaluation of a forensic technique by using this tool, users need to provide the detection function, configure the profile of image datasets and run the test functions.

Note that most forensic techniques use predefined thresholds, which influence the detection results. In our approach, the analyzed techniques cannot be changed by the test tool. The tool works only with hard decision detectors, which generate a binary output whether the detected image is forged or original. In order to get soft decisions as well as showing the mutual relationship between different parameters, users can try different thresholds and apply the framework to several versions of the forensic technique.

3.3 Summary

In this chapter, we proposed a test framework for the evaluation of digital image forgery detection techniques. With the framework, the techniques can be assessed by using common metrics, datasets and attacks in order to measure their effectiveness and security. Since the techniques were tested under the same condition, they can be compared in a fair manner. We built a test tool to support the evaluation process of image forgery detection techniques easily and automatically.

4 Security of Copy-Move Forgery Detection Techniques

4.1 Introduction

In this chapter we first survey techniques for copy-move forgery detection techniques for digital images and propose a method to evaluate them. We discuss several widely-used techniques, implement them and evaluate them by using the test framework proposed in Chapter 3. Subsequently, experimental results allow assessing the effectiveness, robustness, security, and image perceptual quality of the considered techniques. In order to evaluate the security of these techniques, we design different targeted attacks against each of them. Finally, we propose a new technique, which has higher robustness against some common attacks, such as rotation or Gaussian noise addition.

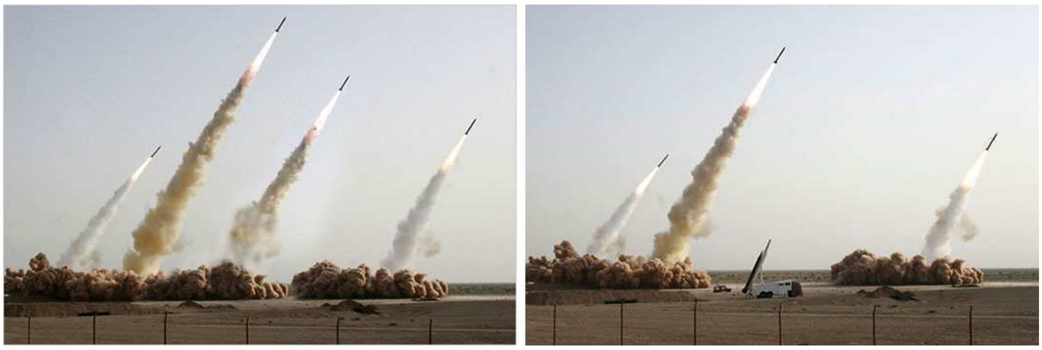


Figure 4-1: Shown on the left is the copy-moved image and shown on the right is the original image.

Detection of copy-move forgeries is a popular image forensic problem, for which many forensic techniques have been developed. The purpose of a copy-move forgery is mostly to hide an important object of an image by covering it with a part copied from another region within the same image. When it is done skillfully, it is difficult to detect by human eyes. Moreover, because the copied parts come from the same image, most important statistical properties of the copied parts are similar to the rest of the image and thus it will be difficult to detect forgeries by using methods that look for incompatibilities in statistical measures in different parts of the image [18]. An example of this forgery can be seen in Figure 4-1, which shows the image of an Iranian missile test that appeared on the front page of many newspapers. However, it was revealed later that the second missile from the right was copied and moved from the third missile in order to conceal the fact that a missile on the ground did not fire.

Many techniques for copy-move forgery detection have been proposed in the literature. In order to detect duplicated regions in an image, a simple approach is performing an exhaustive search, in which the image is compared with its circularly shifted versions. Since that would examine every possible pair of image regions, this method is computationally very demanding [18]. Most existing techniques for copy-move forgery detection follow a more efficient approach based on small fixed-sized overlapping image blocks. The detection process of this approach is presented in Figure 4-2. The analyzed image is firstly divided into overlapping blocks. Specific features of each block are extracted in a succinct way in order to reduce computational complexity as well as increase robustness of the techniques. Although it is not required, most techniques use a subsequent sorting step in order to reduce the complexity for matching similar regions. To this end, the features of each block are usually vectorized, then these vectors are sorted and all pairs of contiguous vectors are examined in order to identify similar blocks. Finally, to judge if the image is forged, in the matching step, one searches for two groups of connected blocks so that every pair of similar blocks has the same distance in the image.

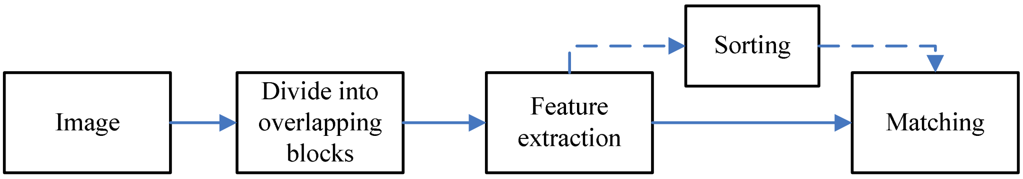


Figure 4-2: The general scheme for detecting copy-move forgeries in digital images.

The most distinguishing property of various copy-move forgery detection algorithms is the employed extracted features. The feature extraction methods of most existing detection techniques can be categorized into four main groups: techniques based on frequency transformation, dimensionality reduction, moments and color features [75]. The features can be obtained from the frequency domain by using the Discrete Cosine Transform (DCT) [18], Discrete Wavelet Transform (DWT) [83] or Fourier-Mellin Transform (FMT) [34]. In order to reduce dimensions of block features, the technique in [35] uses the Principal Component Analysis (PCA) while the technique in [84] uses Singular Value Decomposition (SVD). A typical technique based on color features is proposed in [85]. Some techniques are based on moments, such as [86] uses blur moments and [87] uses Zernike moments.

An important problem in copy-move forgery detection is to find a robust representation for the image blocks, so that duplicated blocks can de identified even after modifications have been applied to the forged image [73]. The authors in [88] proposed rotation-invariant features by using log-polar transform. Another rotation-invariant selection method called Same Affine

Transformation Selection (SATS) is presented in [89]. Some other special feature descriptions such as the Scale Invariant Features Transform (SIFT) [90] and Speed-Up Robust Feature (SURF) [91] have been employed as well. Several SIFT-based techniques for copy-move forgery detection have been proposed by Huang et al. [92], Pan and Lyu [93], and Amerini et al. [94]. The authors in [95] applied SURF features in their technique.

Since many copy-move forgery detection techniques have been proposed, it is important to evaluate and compare them. There are some existing works that compare different detection schemes. The authors of [73] evaluated effectiveness and robustness against some geometric transformation of three techniques based on DCT [18], PCA [35] and FMT [34]. They used only a few images in their test and performed no security and perceptual quality evaluation. The work in [75] presents a common pipeline for copy-move forgery detection and performs a comparative study on a number of detection techniques. The authors introduce a benchmark database for evaluation of copy-move forgery detection techniques. They tested the robustness of the techniques only against two geometric transformations (scaling and rotation), and a perceptual quality measurement is also not included.

4.2 Effectiveness Analysis

4.2.1 Implementation of Detection Techniques

A major obstacle when evaluating detection techniques is that their implementations are often not available. Therefore, we first implemented selected algorithms based on the short descriptions in the papers [18], [35], [85]. In this section, we briefly review these techniques and give some notes on the implementation. All techniques follow the general scheme for copy-move forgery detection presented in Figure 4-2. Firstly, the analyzed $M \times N$ image is divided into overlapping blocks of size $B \times B$ pixels, resulting in $(M–B+1) \times (N–B+1)$ image blocks. Next, the characteristic features of every overlapping block are extracted, and then the features are vectorized. Finally, these vectors are sorted lexicographically in order to identify similar blocks.

Fridrich et al. [18] used several low quantized frequency DCT coefficients for feature extraction. There is no specific information on the number of DCT coefficients given in the paper. In our implementation of [18], we use only six lowest DCT coefficients in each block. In the lossy JPEG compression process, higher quantized DCT coefficients will be eliminated, therefore, the extracted features in [18] are expected to be robust against JPEG compression. In order to prevent too many false matches, the authors of [18] used a large block size, $B = 16$.

They also computed a new 16×16 quantization table based on the 8×8 standard quantization table and an experimental formula.

Similarly to the technique of Fridrich et al. [18], but instead of using DCT, Popescu and Farid [35] performed PCA for every overlapping 8×8 image block to produce a short representation; truncation of the PCA basis reduces the number of dimensions [35]. This representation is also known to be robust to minor variations in the image due to additive noise and lossy compression.

Luo et al. [85] proposes a method to use features based on the information in color channels. Following the method, in each image block, two groups of features are calculated: 1) the first three features (c_1, c_2, c_3) are the averages of red, green, blue components respectively; and 2) The Y channel (Y = 0.299R + 0.587G + 0.114B) is divided into 2 equal parts in 4 directions, then the last four features are computed: c_i = sum(part$_1$) / (sum(part$_1$) + sum(part$_2$)), where i ranges over the image partition depicted in Figure 4-3. Since we use gray-scale images in our experiments and in order to use the same datasets with other considered techniques, we modified the algorithm [85] to deal with gray-scale images. Most details of the algorithm are preserved, except the transition from three color channels to only one channel. That may be the reason that our experimental results of the modified scheme are not fully comparable to the original paper [85].

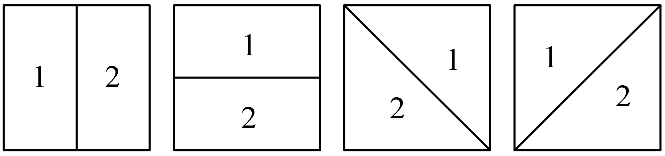


Figure 4-3: Image regions used in the method by Luo et al. [85].

An image is declared to be as forged if there are two groups of connected blocks, where each block in a group has a corresponding similar block in the other group, and every pair of similar blocks has the same spatial distance in the image. To reduce false matches, an image is considered as forged if and only if the number of similar pairs is greater than a predefined threshold. Larger values of the threshold may cause the algorithm to miss some not-so-closely matching blocks, while too small values increase false positives [18]. In each technique, this threshold is usually determined through experiments.

All forensic techniques are based on some predefined thresholds. Although we tried to follow exactly the original papers, the thresholds of the analyzed techniques were adjusted slightly to get optimal results with our dataset (e.g. the detection rate of forged dataset is high while obtaining low false positive). In our experiment we select the thresholds for all tested techniques so that the detection rate is larger than 95% in a test with the datasets of forged images and the false positive rate is lower than 5% on a test with the dataset of original images.

4.2.2 Effectiveness Analysis of Detection Techniques

We apply the test framework to empirically evaluate the three schemes [18], [35], [85]. We create different datasets of original images, forged images and attacked images following the framework of Chapter 3. As original images we randomly selected 200 uncompressed images from [82] and then converted them to gray-scale. The dataset of forged images is created by randomly copying a part in each original image and moving it to another position of the same image. The copied parts can be square or non-square. In the first case, we set the size of the copied parts to 64×64 pixels. In the second case, in order to make it easier for automatic tests, all copied parts are created in the same way by taking in each image a square region of size 64×64, but eliminating the two small square parts of size 12×12 in its upper-left and bottom-right corners. Experimental results are shown in Table 4-1 and Table 4-2. In general, all analyzed techniques can detect forgeries with high accuracy. In the case of square copied parts, all techniques work well with detection rates (DR) higher than 95% and correctness rates (CR) higher or equal to 94%, while the incorrectness rates (ICR) are very low. For non-square parts, there are no significant differences in comparison to the previous case, except [18] which has a lower *CR*. The truncating and rounding processes of the technique [18] are the causes for its lower detection rate; however, these processes make [18] more robust against some post-processing operations.

	Fridrich et al. [18]	Popescu and Farid [35]	Luo et al. [85]
DR	95.5%	100%	99.5%
CR	94.0%	99.5%	99.0%
ICR	1.5%	0.5%	0.5%

Table 4-1: Detection rates, correctness rates and incorrectness rates while evaluating forged images where copied parts are square.

	Fridrich et al. [18]	Popescu and Farid [35]	Luo et al. [85]
DR	94.5%	100%	100%
CR	80.0%	99.5%	99.0%
ICR	14.5%	0.5%	1.0%

Table 4-2: Detection rates, correctness rates and incorrectness rates while evaluating forged dataset where copied parts are non-square.

4.3 Robustness and Security Analysis

In the papers [18], [35], [85], the authors made some robustness tests, but did not perform any security evaluation. In this section, we analyze the robustness as well as the security of the above techniques under identical conditions. Robustness can be assessed by applying the detection techniques to a set of forged images, which underwent post-processing operations. Several different post-processing operations were listed in Section 3.2.4. Furthermore, we also design dedicated attacks in order to evaluate the security of the detection techniques.

At the beginning, we tested the robustness of the techniques under Gaussian noise addition of various SNR of 24, 29, and 40 dB. The experimental results are shown in Table 4-3. Although [18] did not show any robustness test, our tests indicate that the DCT-based technique is extremely robust against Gaussian noise addition. The techniques of Popescu and Farid [35] and Luo et al. [85] show less robustness.

The robustness against JPEG compression with different quality factors (QF) of 40, 60, and 90 is shown in Table 4-4. Since [18] is a DCT-based technique, it is mostly robust against JPEG compression with correctness rates about 90%, even in the case of a low quality factor of 40. The technique of Popescu and Farid [35] is quite robust only in the case of a high compression quality factor. The pixel-based technique [85] has very high false positive rate and it is mostly defeated with a correctness rate of only 1.5%.

Lastly, we examined the robustness against image rotation with the angles of 1, 2, and 3 degrees (Table 4-5). Although [18] seems not so robust against rotation, with correctness rates of about 60% in the tested cases, it is the best amid the analyzed techniques. The PCA-based technique of Popescu and Farid [35] is not very robust against geometric transformations, because the re-sampling operations in the transformation process affect the eigenvalues. The modified technique of Luo et al. [85] is based on pixel values in the spatial domain, so it is highly sensitive to any geometric transformation.

SNR	Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
	DR	CR	DR	CR	DR	CR
24 dB	83.5%	80.5%	15.0%	15.0%	0%	0%
29 dB	92.5%	91.0%	46.5%	46.5%	0%	0%
40 dB	94.5%	93.0%	89.0%	89.0%	1%	0%

Table 4-3: Detection rates and correctness rates after Gaussian noise addition.

QF	Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
	DR	CR	DR	CR	DR	CR
40	90.0%	89.0%	43.5%	11.5%	32.0%	1.5%
60	93.0%	92.0%	45.5%	16.0%	23.0%	1.5%
90	95.0%	93.5%	79.0%	78.0%	3.5%	1.5%

Table 4-4: Detection rates and correctness rates after JPEG compression.

Angle	Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
	DR	CR	DR	CR	DR	CR
1°	62.5%	60.0%	25.0%	25.0%	1.5%	1.5%
2°	72.0%	65.0%	33.0%	26.5%	1.0%	0%
3°	77.0%	61.0%	63.0%	24.0%	2.0%	0%

Table 4-5: Detection rates and correctness rates after rotation.

After checking the robustness with respect to common image processing operations, we report results for some targeted attacks that use specific characteristics of the detection techniques.

Attack 1: We considered a simple operation which is widely-used in watermarking. This assigns the least significant bit (LSB) of each pixel a random value in {0, 1}. The operation can change every image pixel but the perceptual quality of the image is mostly not affected. The results (in Table 4-6) show that the DCT-based technique and PCA-based technique are robust against this attack, while the color pixel-based technique by Luo et al. [85] is completely defeated.

Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
DR	CR	DR	CR	DR	CR
94.5%	92.5%	89.0%	89.0%	1.0%	0%

Table 4-6: Detection rates and correctness rates after changing LSB.

Attack 2: This attack is a combination of simple geometric transformations. Though geometric transformations often distort images, they may be effective against many copy-move forgery detection methods. In this attack, a small portion of the image is cropped and subsequently the cropped image is rescaled to its original size in order to hide trace of the transformations. Given a forged image of size $M \times N$, we crop S pixels so that the cropped image is the rectangle part $[S, S, M–S, N–S]$ of the forged image. Then the cropped image is rescaled to the previous size using bi-cubic interpolation. Through some experiments on different S values, we found that when S is equal to 3, the attack is more powerful, defeating the techniques of Popescu and

Farid and Luo et al. and affecting the technique of Fridrich et al. in 50% of the cases (see Table 4-7).

Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
DR	CR	DR	CR	DR	CR
50.0%	44.5%	5.5%	4.5%	4.0%	2.5%

Table 4-7: Detection rates and correctness rates in evaluation for cropping and rescaling.

Attack 3: Since the technique [18] is DCT-based, we consider attacks that directly modify DCT coefficients. A possible attack works similar to a watermarking scheme proposed by Koch and Zhao [96] which operates on DCT coefficients. Firstly, the image is divided into overlapping blocks of size 16×16 pixels, and then each block is transformed by DCT. We choose two random DCT coefficients among the lowest AC coefficients. These two coefficients are swapped and a random small positive number is added to one of them. At the end, all DCT blocks are transformed back from the frequency domain into the spatial space by the Inverse Discrete Cosine Transform (IDCT). Experimental results are shown in Table 4-8. Although the DCT-based technique [18] is very robust against JPEG compression, it is not robust to this attack, because the attack directly modifies DCT coefficients. However, the disadvantage of this attack is its low fidelity (i.e. the quality of attacked images is reduced), especially when the image has large homogeneous regions.

Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
DR	CR	DR	CR	DR	CR
42.5%	36.0%	92.5%	83.5%	9.0%	4.5%

Table 4-8: Detection rates and correctness rates after swapping DCT coefficients.

Attack 4: Through the above tests, we found that skillful geometric transformation operations can be very effective attacks against some of the detection techniques. Although a direct attack on DCT coefficients can be effective against the technique of Fridrich et al., it resulted in rather low fidelity. For these reasons, we choose another alternative attack where we use cropping, rescaling, and JPEG compression, instead of directly manipulating DCT coefficients. Firstly, the forged image is cropped by 3 pixels, and then the cropped image is JPEG compressed two times with different qualities, 70 and 60. The image is then converted to the original format and rescaled to the original size. The effectiveness of this attack is quite impressive, while its fidelity is very high. According to the experimental result is shown in Table 4-9, only about 30% of forged images are detected by [18] and the other techniques are mostly defeated.

Fridrich et al. [18]		Popescu and Farid [35]		Luo et al. [85]	
DR	CR	DR	CR	DR	CR
35.0%	31.0%	23.0%	3.5%	12.5%	0%

Table 4-9: Detection rates and correctness rates after the combination attack of cropping, double JPEG compression and rescaling.

Visual quality degradation due to the attack is an important issue to be considered in order to develop a good attack. Using the same image datasets, which are used to evaluate the effectiveness and security of the detection techniques, we calculate the PSNR and WPSNR between forged images and their attacked versions. We take the average PSNR and WPSNR of every pair images from the dataset of forged images and the dataset of attacked images. The results are shown in Table 4-10, Table 4-11, and Table 4-12. Although geometric transformations usually degrade the visual quality, in the case of cropping with a small number of pixels and rescaling to the previous size, it is also difficult to realize the manipulations. Thus, our proposed targeted attacks are still useful.

SNR	PSNR	WPSNR
24 dB	20.46 dB	33.41 dB
29 dB	22.86 dB	36.42 dB
40 dB	28.41 dB	42.94 dB

Table 4-10: Visual quality evaluation of attacked images by adding noise.

QF	PSNR	WPSNR
40	20.08 dB	33.06 dB
60	20.99 dB	34.35 dB
90	24.63 dB	38.87 dB

Table 4-11: Visual quality evaluation of attacked images by JPEG compressing.

	PSNR	WPSNR
Swapping DCT coefficients	23.37 dB	34.70 dB
Cropping and rescaling	14.79 dB	24.74 dB
Cropping, compressing and rescaling	14.69 dB	24.65 dB

Table 4-12: Visual quality evaluation of several attacks.

4.4 A New Technique for Copy-Move Forgery Detection

An important characteristic for detection techniques is to use a robust representation, so that duplicated blocks can be identified from a forged image even if the image was post-processed. The technique of Fridrich et al. [18] can be robust against some image modifications, but cannot resist more specific attacks as shown in the last section.

In this section, we design a new technique for copy-move forgery detection. In the proposed technique, we use the Radon transform for extracting block features and the phase correlation for matching similar blocks. Through our evaluation, we show that the proposed technique is more robust against some image post-processing operations, such as rotation and Gaussian noise addition, than the technique of Fridrich et al. [18]. We also realize that our technique is more robust than [18] when in-processing attacks are applied, i.e. where a part of an image is rotated before it is moved to a different place in the image.

4.4.1 Radon Transformation and Phase Correlation

A. Radon Transformation

The Radon transformation computes projections of an image along the directions given by various angles, as shown in Figure 4-4 (where r is the perpendicular distance of a line from the origin and θ is the angle formed by the distance vector). The result of the Radon transforms of an image $f(x, y)$, denoted as $g(r, \theta)$, and is the sum of the intensities of the pixels in each direction, i.e. a line integral. It is possible to express the Radon transformation as follows:

$$g(r,\theta) = R(f(x,y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\delta(r - x\cos\theta - y\sin\theta)\,dxdy \,,$$

where we used the sifting property of the *impulse function* δ. This function reduces the double integral to a projection beam in the direction θ that has a distance r from the center of the coordinate system [97]. The Radon transformation has robustness properties against rotation, scaling, and translation (RST) operations [98], [99] and it is also robust against additive noise [100].

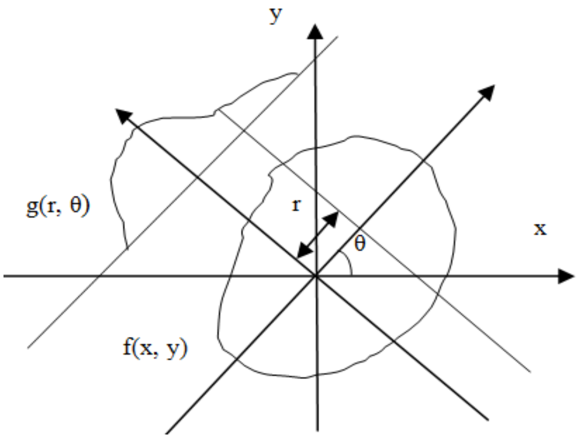


Figure 4-4: The Radon transform of *f(x, y)* projects follows a particular direction.

B. Phase Correlation

The phase correlation is a technique to measure the similarity of two images (or image blocks) of the same size. To compute the maximum phase correlation of two images I_1 and I_2, one performs in following steps:

1. Apply the Fourier transform F on both images:

$$F_1 = F(I_1),$$

$$F_2 = F(I_2).$$

2. Calculate the cross-power spectrum by element-wise multiplying the first result F_1 with the complex conjugate of the second result F_2 and normalizing this product:

$$R = \frac{F_1.F_2^*}{|F_1.F_2^*|} ,$$

where x^* denote the complex conjugate of and $|\, x \,|$ is the complex magnitude of x.

3. Obtain the normalized cross-correlation by applying the inverse Fourier transform F^{-1} to the cross-power spectrum:

$$IR = |F^{-1}(R)| .$$

4. Compute the maximum of the phase correlation:

$$PC = max\,(IR) .$$

The maximum phase correlation *PC* infers the location of the peak of the relative translation offset between the pair images I_1 and I_2, and can be used as a similarity criterion.

4.4.2 The Proposed Technique

In this section, following the general scheme presented in Figure 4-2, we propose a new technique based on Radon transformation and phase correlation. The technique consists of the following steps (a detailed description is presented in Algorithm 4-1).

Segmentation: Divide the $M \times N$ image to be tested into overlapping blocks. The image is scanned from the upper left corner to the lower right corner, sliding a $B \times B$ block over the image. This results in $(M–B+1)\times(N–B+1)$ blocks.

Feature extraction: For each block, apply the Radon transform in various directions, which are specified by a set of angles. The result is a matrix where each column contains the projections with respect to predefined angle.

Sorting: Vectorize the transformed matrices of every block and store each of them as a row in a matrix. Subsequently, the rows are sorted lexicographically.

Matching: Compute the maximum phase correlation of two image blocks, which are corresponding to every pair of contiguous rows. Two blocks are approximate if their maximum phase correlation is larger than a predefined threshold.

Decision: If there are two groups of connected approximate blocks, where every pair of approximate blocks has the same spatial distance in the image and the number of blocks in each group is larger than another predefined threshold, we rate the image as forged.

Algorithm 4-1: Improved technique for copy-move forgery detection.

```
1. Input an $M \times N$ image.
2. Initialize the parameters:
       ▪ $B$: fixed size of image block.
       ▪ $\theta$: set of angles (e.g. $\theta = \{0, 1, 2, \ldots, 179\}$).
       ▪ $T_1$: maximum phase correlation threshold (range from 0 to 1).
       ▪ $T_2$: minimum offset threshold.
       ▪ $T_3$: threshold on the number of connected image blocks.
```

- C: a $M \times N$ matrix, which is initialized by zero.

3. Apply Radon transform for a set of angles θ to each overlapping $B \times B$ block, resulting in a matrix R (each column of R is the Radon transform for one of the angles in θ).
4. Vectorize R for each image block and store it as a row of a matrix S. Subsequently, the number of rows of S corresponds to the same of the number of image blocks.
5. Sort the rows of S in lexicographic order. Let $\vec{s}_i$ denotes the row i of S; let b_i denote the block corresponding to $\vec{s}_i$ (note that unlike $\vec{s}_i$ and $\vec{s}_{i+1}$, the blocks b_i and b_{i+1} are usually not adjacent); let (x_i, y_i) denote the position of the block b_i in the image (top-left corner).
6. For every pair of $\vec{s}_i$ and $\vec{s}_{i+1}$
    - Compute the maximum of phase correlation between the b_i and b_{i+1}, resulting in PC_i.
    - If $PC_i > T_1$ then

```
      If $(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 > T_2$ then
          $u = |x_i - x_{i+1}|$
          $v = |y_i - y_{i+1}|$
          $C(u, v) = C(u, v) + 1$
      End if
  End if
```

7. If $C(u, v) > T_3$ for any pair (u, v) then the image is judged as forged.

4.4.3 Experimental Results

In this section we evaluate our proposed technique (called RTPC) and compare to the DCT-based technique [18] (called DCT). We use the framework mentioned in Chapter 3 for this purpose.

In the Radon transform, the more angles we use the cost in terms of time and memory consumption increases. In our experiments, we specify projection directions by angles from 0° to 179° with a reasonable incremental step of 10°. To choose a good threshold for the maximum phase correlation, we test the RTPC technique on several image datasets by using different values of the threshold (T_1) of 0.7, 0.8 and 0.9. We found that once the threshold T_2 is fixed to 20, the detection rates in detecting the forged dataset are always larger than 99%, but the false positive rates in classifying original images are 18%, 12% and 5.5% respectively. Therefore, in our experiments, we set the threshold $T_1 = 0.9$. With the same parameters of the threshold $T_2 = 20$, the technique [18] has a detection rate of 99.5% in detection of the forged dataset and a false positive rate of 4.5% in detection of the original dataset. The results show that both techniques work well in the absence of attacks.

Next, we test the techniques against some common attacks such as rotation and noise addition. The experimental results are presented in Table 4-13 and Table 4-14. We realize that both analyzed techniques are quite robust against rotation with the small angle of 1°. While the technique [18] is not robust against rotation with larger degrees, the RTPC technique is more robust and can detect rotated images by 3° with detection rates larger than 70%. While [18] seems robust against Gaussian noise addition only with the SNR larger than 40 dB, the RTPC technique is more robust against noise addition in most cases.

Angle	1°	2°	3°	4°
DCT [18]	94.0%	65.0%	41.5%	26.5%
RTPC	99.5%	96.5%	71.0%	36.0%

Table 4-13: Detection rates for detecting rotated images ($B = 16$).

SNR	45 dB	40 dB	35 dB	30 dB
DCT [18]	96.0%	46.0%	15.0%	5.0%
RTPC	94.5%	88.5%	67.5%	29.0%

Table 4-14: Detection rates for detecting forged images with added Gaussian noise ($B = 16$).

Angle	1°	2°	3°	4°
DCT [18]	36.0%	23.5%	20.5%	15.5%
RTPC	72.0%	28.0%	25.0%	19.5%

Table 4-15: Detection rates for detecting copy-rotate-moved images ($B = 16$).

Additionally, we test the techniques with a simple in-processing attack. The dataset of attacked images is created by copying a random square part of size of 64×64 pixels in each image from the original dataset, rotating the copied part with a small angle, and then pasting it at another position of the same image. The experimental results are shown in Table 4-15. We can see that the technique [18] is not robust against this attack and the RTPC based technique is rather robust only in the case of the small rotation angle of 1°.

In order to keep the compatibility with the paper [18], in the previous tests, we used an image block size of 16×16 pixels (B = 16) for both tested techniques. However, in copy-move forgery detection, the block size can affect the detection rates of the forensic techniques significantly. Thus, we evaluate the techniques with another block size of 8×8 pixels (B = 8). We set the threshold T_2 to 25 in order to keep the false positive of the technique at 4.5% while the detection rate at 99.5%. Since the original technique of [18] uses the block size of 16, some thresholds of the technique [18] were adjusted so that the false positive is lower than 5% and the detection rate in the absence of any post-processing or attack is about 99%.

The experimental results of the detection techniques when B = 8 in the tests of rotation, Gaussian noise addition and an in-processing attack are shown in Table 4-16, 4-17, and 4-18. We realize that by using a smaller block size of 8×8 pixels, the proposed technique is robust against rotation with angles smaller than 4°, Gaussian noise addition with SNR larger than 35 dB and the in-processing rotation with an angle of 1°. In comparison to [18], our technique is more robust in all test cases.

Angle	1°	2°	3°	4°
DCT [18]	91.0%	52.5%	34.5%	23.0%
RTPC	99.5%	99.5%	79.0%	49.5%

Table 4-16: Detection rates for detecting rotated images (B = 8).

SNR	45 dB	40 dB	35 dB	30 dB
DCT [18]	35.0%	3.5%	0%	0%
RTPC	92.0%	89.0%	67.5%	25.0%

Table 4-17: Detection rates for detecting images with added Gaussian noise (B = 8).

Angle	1°	2°	3°	4°
DCT [18]	41.5%	31.5%	28.0%	19.5%
RTPC	82.0%	47.5%	40.5%	23.0%

Table 4-18 Detection rates for detecting copy-rotated-moved images (B = 8).

4.5 Summary

Copy-move forgery detection is an important problem in the field of digital image forensics. In this chapter we applied the proposed framework to evaluate the effectiveness, robustness and security of three copy-move forgery detection techniques of Fridrich et al. [18], Popescu and Farid [35] and Luo et al. [85]. The effectiveness and security of the techniques were evaluated by examination of their detection rates and correctness rates in tests on forged images and attacked images respectively. We designed some targeted attacks based on the analysis of the characteristics of each technique and thereby assessed their security. We have shown that all tested techniques can effectively be defeated by rather simple signal processing techniques. It is then possible to disguise a forged image as authentic. Lastly in this chapter, we designed a new technique for the detection of copy-move forgeries. The technique uses the Radon transform and phase correlation, resulting in more robust results in comparison with the baseline technique [18] in the same test conditions.

5 Security of Re-sampling Detection Techniques

5.1 Introduction

In order to create convincing forged images, one usually applies geometric transformations to the images or to a part of them. Many geometric transformations such as rotation and resizing involve a re-sampling step, which may not be easily realized by human eyes. Interpolation is the central step of re-sampling in order to estimate the value of a signal at intermediate positions of the original samples. This step is the key to smooth the signal and create a visually appealing image [101]. Since interpolation creates specific statistic artifacts in the re-sampled images, detecting traces of re-sampling is a popular approach in the field of image forensics.

Many re-sampling detection techniques have been proposed, and they can be divided into two main approaches. The first approach is based on predicting the dependency of neighboring image pixels [38], [102], [103]. The second approach is based on computing the variance of the second derivatives of the analyzed image [40], [41], [104]. To evaluate the effectiveness of the techniques the authors of [38] used 200 uncompressed images as the original dataset, and selected 50 images to create re-sampled versions. The number of testing images in [102] and [103] is 200 and all of them have been used to produce re-sampled images. The techniques in [104] were tested with only one image, while in [40] used 114 images, and [41] used 40 images. To evaluate the robustness of a certain technique, the authors usually apply several post-processing operations to re-sampled images.

It is obviously difficult to judge which detection technique is better since they were evaluated on different datasets under different testing conditions. To fill this gap, Uccheddu et al. [105] proposed an experimental methodology and applied it to evaluate and compare the two re-sampling detection techniques of Kirchner and Gloe [103] and Mahdian and Saic [41]. In the paper, the authors used a dataset of 200 images in different categories; both analyzed techniques were tested following the same methodology. In the effectiveness test they considered only re-scaled images and in the robustness test they limited their study to JPEG compression.

In this chapter, we study three well-known re-sampling detection techniques of Gallagher [40], Mahdian and Saic [41], and Popescu and Farid [38]. Next, we apply the general test framework of Chapter 3 to evaluate the effectiveness and robustness of the techniques. We design some targeted attacks in order to defeat the techniques. Consequently, we can assess the security of the detection techniques.

Lastly, we propose some improvements which enhance the robustness despite post-processing operations.

5.2 Effectiveness Analysis

5.2.1 Implementation of Detection Techniques

Since the implementation of re-sampling detection techniques are often not available, we first implemented detection algorithms according to the description in published papers. In this section, we review the techniques in [38], [40], [41] and give details on their implementations.

Gallagher realized that low-order interpolated signals introduce periodicity in the variance of their second derivatives with a period that is equal to the re-sampling factor [40]. This observation can be used to detect whether an image has been re-sampled. Specifically, the periodicity is uncovered by computing the discrete Fourier transform (DFT) of the second derivatives of the analyzed signals. In image forensics, the signals are rows (or columns) of the analyzed image.

Although Mahdian and Saic [41] proved more generally that the variance of the n^{th} derivative of a re-sampled signal is also periodic, they used only the second derivative in their experiments. This detection algorithm consists of the following steps. Firstly, in a similar way as [40], the second derivatives of the analyzed signal is calculated. Next, the Radon transformation (see Section 4.5.1) is employed to compute projections of magnitudes of the second derivatives along specified directions. The authors apply this algorithm to every row (or column) of the examined image. The implementation of the core part of the technique is available on the website of the authors [106].

The algorithm of Popescu and Farid [38] is probably the most widely used method. The authors noted that there are linear dependencies between neighboring image samples (pixels) in re-sampled images. In order to determine these correlations, they employed the expectation/maximization (EM) algorithm [107] to estimate the linear correlation between each pixel and its neighbors, eventually computing the probability of each sample being correlated to its neighbors. To this end, the technique employs a linear predictor to approximate the value of each sample y_i as the weighted sum of its surrounding $N\times N$ samples. Thus, the residue of each sample y_i and its neighbors can be modeled as:

$$r_i = y_i - \sum_{k=-N}^{N} \alpha_k y_{i+k} \quad .$$

The correlation probability p_i of each sample is computed based on the prediction error r_i which is modelled as a zero-mean Gaussian random variable:

$$p_i = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-r_i^2}{2\sigma^2}\right) .$$

The probability values of all samples of an image together form the probability matrix (called p-map). The authors of [38] empirically found that the p-map of a re-sampled image is periodic and the periodicity becomes evident through the peaks in the frequency domain. However, the values of the weights (α) are usually not known in practice, so the p-map can not be computed directly. The authors of [38] use an initial set of α for the estimation and then use Weighted Least Squares (WLS) integrated into an iterative EM algorithm in order to optimize the values of α and estimate the correlation of neighboring samples.

The detection results are transformed to the frequency domain in order to uncover interpolation artifacts in the form of peaks. To quantify the performance of these techniques, we use a threshold-based peak detector that reaches for local maxima (peaks) in the frequency domain. Since there is a trade-off between the detection rate and the false positive rate (FPR), the threshold has been chosen carefully through experiments. In tests with the framework proposed in Chapter 3, we found that the techniques of Gallagher [40] and Mahdian and Saic [41] have a high FPR: when we adjust thresholds so that their detection rates (in test the dataset of forged images) are larger than 90%, their FPR (in test the dataset of original images) is lower than 18%. At the same time, the detection rate of [38] is larger than 90% while its FPR is rather low (about 6%). As an effort to reduce the FPR for the techniques in [40], [41] to below 10% by adjusting the thresholds which they used, we found that their detection rates decreased significantly, so we missed many forgeries. The reason for the higher FPR is that many false positives were caused by strong textures. Since the techniques in [40], [41] are based on examining the second derivatives of images, strong textures produce periodic patterns in original images, which yield peaks in the frequency spectrum similar to re-sampled images.

5.2.2 Effectiveness Analysis of Detection Techniques

In this section, we apply the test framework to assess the effectiveness of re-sampling detection techniques of Popescu and Farid [38], Gallagher [40] and Mahdian and Saic [41]. Firstly, we create a dataset of original images by randomly collecting 200 uncompressed images from [82]. The images are then converted to gray-scale, and cropped to the size of 256×256 pixels. We created different datasets of up-sampled, down-sampled, and rotated images with different factors. Since all of these techniques employ statistical methods, their effectiveness can be affected by the dataset in use. Thus, we also evaluate the techniques on a dataset of 128×128 images and 256×256 images.

All re-sampled images are created from the original image dataset by using the *imresize* function of Matlab. We found that the techniques can detect up-sampled images with scaling factors are larger than 1.1 rather well. They detect perfectly (with a detection rate of almost 100%) re-sampled images of size 128×128 pixels by a scaling factor larger than 1.3, and re-sampled images of size 256×256 by a scaling factor larger than 1.2. Gallagher [40] showed that in the special case of interpolation by a factor of 2.0, there are no meaningful peaks produced in normalized frequency. This is confirmed by our experiments.

The experimental results for detecting up-sampled images of size 128×128 and 256×256 pixels are presented in Figure 5-1 and Figure 5-2. Since the techniques are based on statistical methods, using larger images for testing, we apparently get stronger and more accurate detection results. In the same way of testing up-sampled images, we tested down-sampled images with different scaling factors from 0.4 to 0.9. We realized that the detection rates of the techniques in detecting down-sampled images are low (Figure 5-3). The reason is that down-sampling causes loss of information, thereby limiting the detection capabilities of the statistical based detection techniques [38], [40], [41].

Following the tests with up-sampled and down-sampled images, we evaluate the detection techniques on rotated images with different angles. All rotated images were created from the dataset of 256×256 original image by using the *imrotate* function of Matlab with bicubic interpolation. In order to reject the black parts in the corners of the rotated images, we crop the image and keep only the center part of size 196×196 of each rotated image for evaluation. We realize that the technique [38] can detect rotated images (with a rotation angle larger than 5 degrees) with a detection rate of about 80%, while the techniques based on investigating the second derivatives of images [40], [41] are not robust against rotation (Figure 5-4).

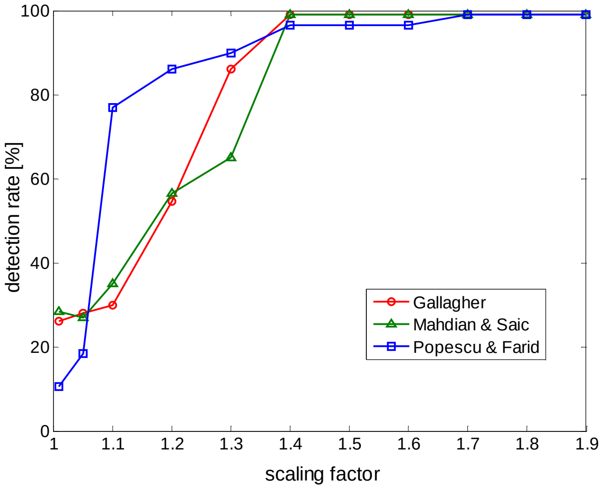


Figure 5-1: Detection rate for 128×128 up-sampled images.

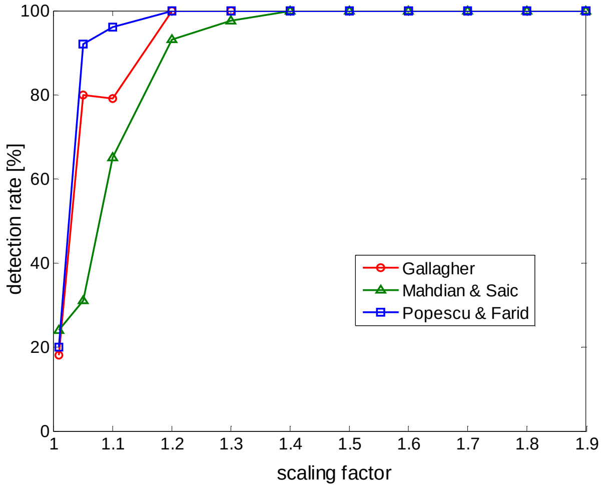


Figure 5-2: Detection rate for 256×256 up-sampled images.

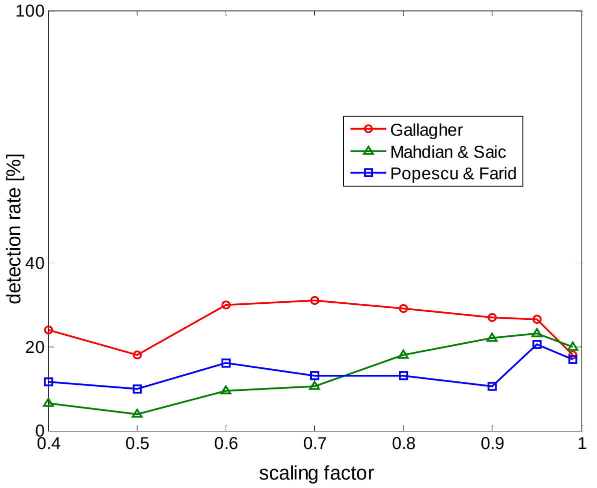


Figure 5-3: Detection rate for 256×256 down-sampled images.

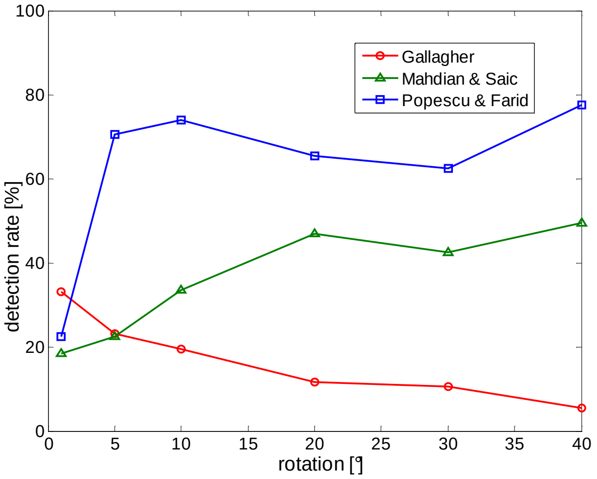


Figure 5-4: Detection rate for 256×256 rotated images.

5.3 Robustness and Security Analysis

5.3.1 Robustness Analysis

To make tampering more convincing, post-processing is commonly applied to re-sampled images. However, post-processing often worsens the effectiveness of detection techniques. In order to assess the robustness of the techniques, we employ different post-processing operations in the re-sampled images. We choose 200 up-sampled images with the factor of 1.2, where the detection rate was very high for all considered techniques. Specifically, the detection rates of the techniques for the up-sampled images with the scaling factor of 1.2 are larger than 90% in the case of images with 256×256 pixels (Figure 5-2).

We applied some post-processing operations such as Gaussian noise addition and JPEG compression to the up-sampled images. The detection results after the post-processing are presented in Figure 5-5 and Figure 5-6. While the techniques of Popescu and Farid [38] and Gallagher [40] are defeated in case of added Gaussian noise with a SNR lower than 30 dB, the technique Mahdian and Saic [41] is more robust. All techniques are more robust against adding Gaussian noise with higher SNR. The technique [40] is sensitive to noise, but it is more robust against JPEG compression than the technique [41]. However, JPEG compression creates blocking artifacts, which introduce periodical peaks, which are similar to the impact of interpolation in the frequency domain. These peaks create false positives in the detection results and thus the detection rates sometimes grow inversely proportional to the quality factors when detecting by using the technique [38].

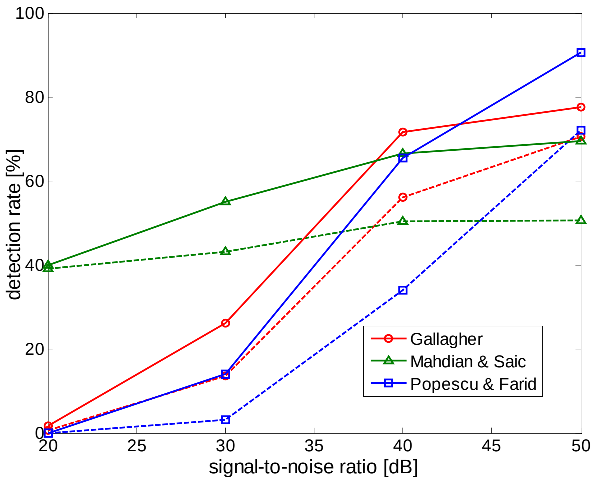


Figure 5-5: Detection rate for up-sampled images with post-processing by adding Gaussian noise (solid lines and dashed lines present results for 256×256 images and 128×128 images respectively).

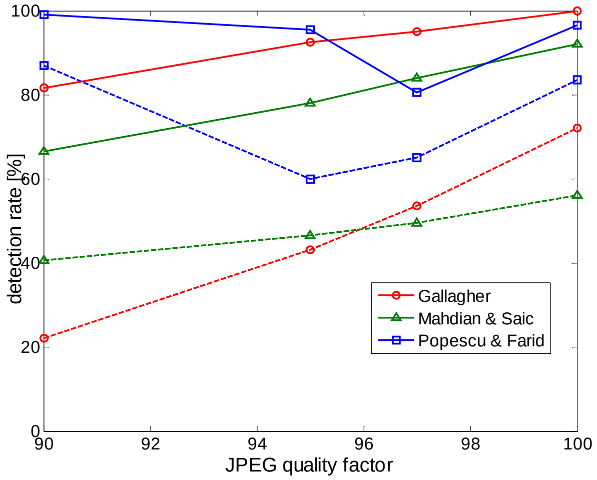


Figure 5-6: Detection rate for up-sampled images with post-processing by JPEG compression (solid lines and dash lines present results for 256×256 images and 128×128 images respectively).

In light of [60], we use median filtering as a post-processing operation to the re-sampled images. Although in the original papers [38], [40], [41], the authors have not considered median filtering, during our experiments, we identify median filtering as an effective attack against several re-sampling detectors. Since the median filter is nonlinear, it defeats well the techniques based on the detection of local linear dependency [38]. The experimental results of the techniques under test with re-sampled images are presented in Figure 5-7. In order to satisfy the trade-off between the attack effectiveness and the quality of the attacked images, we suggest using Gaussian noise with SNR of 30 or median filtering with a window size of 3 (see Table 5-1). Although the techniques of Gallagher [40] and Mahdian and Saic [41] seem more robust against some attacks than the technique of Popescu and Farid [38], we noted that [40], [41] have much higher FPR than [38] with the fixed thresholds we chose for tests.

We propose a new attack by using order-statistic filtering. The filter replaces each pixel in a re-sampled image by the third largest value of the pixel among its north, east, south and west neighbours. We use this filter to attack up-sampled images with a scale factor of 1.2. In Table 5-1, we show the efficiency of the attack against the techniques [38], [40], [41]: it mostly defeats the technique of Popescu and Farid [38].

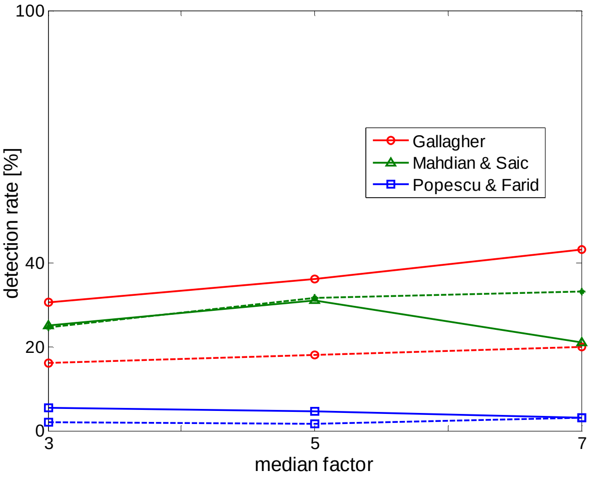


Figure 5-7: Detection rates for up-sampled images with post-processing by median filtering (solid lines and dashed lines present results for 256×256 images and 128×128 images).

A good attack not only reduces the detection rates of detection techniques, but also retains image quality. To quantify this factor of an attack, we compute the average difference between pairs of re-sampled images (before the attack) and attacked re-sampled images (after the attack). The difference is measured by calculating the PSNR: a higher PSNR normally indicates that the attacked image is of higher quality. In Table 5-2 we show the average difference between re-sampled dataset (without any attack) and its attacked versions. It should be noted that, although median filtering is an effective attack to re-sampling detectors, it may leave specific evidence which can be detected and thus reveal the existence of the attack [66].

	Gallagher [40]	Mahdian and Saic [41]	Popescu and Farid [38]
Gaussian noise with SNR=30	26.0%	55.0%	14.0%
Median filter with size=3	30.5%	25.0%	5.5%
Order-statistic filter	71.0%	39.0%	4.0%

Table 5-1: Detection rates for up-sampled images after attack by using Gaussian noise, median filter, and order-statistic filter.

Gaussian noise with SNR = 30	Median filter with size = 3	Order statistic filter
23.3 dB	18.4 dB	21.0 dB

Table 5-2: Average PSNR between re-sampled images and attacked re-sampled images.

The robustness of [38] was determined by applying different countermeasures, such as Gaussian noise addition and JPEG compression to re-sampled images. Nevertheless, the authors of [60] showed that the reliability of the technique was still analyzed only on the surface. Therefore, the authors proposed in [60] some targeted attacks against the technique [38]. The first attack is based on nonlinear filtering, the second attack is based on the Sobel edge detector, and the third attack integrates both mentioned attacks. In the next section, we design some other rather simple but effective targeted attacks against [38]. The first attack is based on multiple re-sampling by specific scales, the second attack is based on hybrid median filtering, and the third attack employs a combination of both. We use the attacks to evaluate the security of our improved technique which we propose in Section 5.4.2.

5.3.2 Attack Based on Multiple Re-sampling

When an image is down-sampled by a factor of two, no sample in the down-sampled image can be written as a linear combination of its neighbors [38]. Subsequently, traces of re-sampling should not be noticed in theory. Hence, we design an attack to disguise a re-sampled image by up-sampling by a factor of two and down-sampling it by a factor of two, thus yielding an image of the original size. We call the process attack by multiple re-sampling.

Figure 5-8 illustrates the detection process of [38] which consists of testing images, their corresponding p-maps and the Fourier transform of the p-maps. We realize that there is no peak in the Fourier transformed p-map of the original image, but in the case of an up-sampled image, its transformed p-map has remarkable peaks. Although the quality of the tested image is not noticeably affected by the attack of multiple re-sampling, at the same time the peaks have not been absolutely eliminated (i.e. the traces of re-sampling can still be uncovered by the re-sampling detector). Using the detector of [38] on a dataset of 200 up-sampled images by a factor of 1.2, we obtained a detection rate of 99%. After applying the attack to the up-sampled images, the detection rate is reduced to 84%.

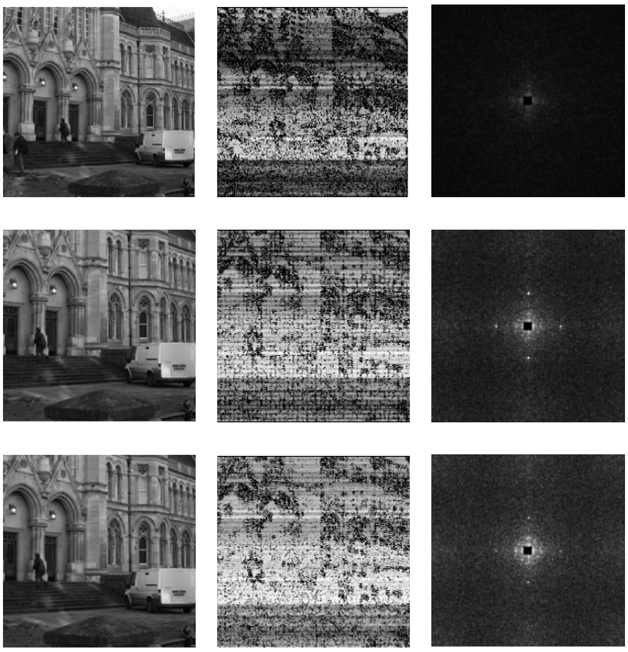


Figure 5-8: Shown in the top row is the original image, in the middle row the same image up-sampled by a scaling factor of 1.2, and in the bottom row the same up-sampled image, post-processed by the attack of multiple re-sampling. Each row shows the image itself, its p-map and the Fourier transform of the p-map.

5.3.3 Attack Based on Hybrid Median Filter

Since the technique [38] is based on detecting linear dependencies between neighboring samples, all kinds of nonlinear filters applied as a post-processing step are candidate attacks [60]. The authors of [60] proposed a targeted attack based on median filtering. While the attack is successful to conceal traces of re-sampling, the visual quality of the attacked images suffers from noticeable blurring. To overcome this drawback, we design a targeted attack which is based on another nonlinear filter called hybrid median filter [108]. The filter consists of three steps, each being applied to a $N \times N$ sliding window (N must be odd). In the first step we compute the median of horizontal and vertical pixels in a $N \times N$ block (called M_1). In the second step we compute the median of diagonal pixels in the block (called M_2). Finally, the filtered pixel value is the median of the two median values (M_1 and M_2) and the center pixel of the block.

Figure 5-9 illustrates the detection results of [38] for both kinds of nonlinear filters. We found that the median filter destroyed most evident peaks in the transformed p-map, but it also makes the image blurry. Conversely, the image attacked by the hybrid median filter is much less blurred, but sometimes peaks are still retained. When testing [38] on a dataset of 200 up-sampled images by a factor of 1.2, the detection rate is 99%. After applying the hybrid median filter to the up-sampled images, the detection rate is degraded to 76%.

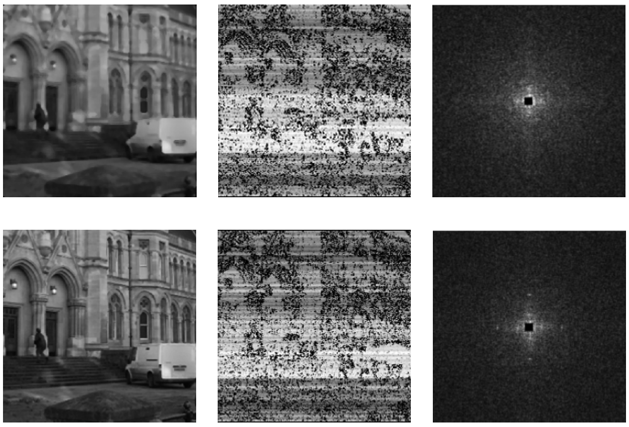


Figure 5-9: Shown in the top row is the up-sampled image attacked by the 3×3 median filter and in the bottom row the same up-sampled image post-processed by a hybrid median filter with N = 3. Again, we show the image, its p-map and the Fourier transform of the p-map.

5.3.4 Combination Attack

Although the proposed targeted attacks reduce the capability of detecting re-sampling, the detection rates are still high. In order to design a more powerful attack, we use them in combination: Firstly, the image is up-sampled by a factor of two, then down-sampled by a factor of two. The image is then anti-aliased in order to remove aliasing artifacts of the down-sampling process. Lastly, a hybrid median filter is applied to the image.

Figure 5-10 illustrates the detection results of an up-sampled image which has been manipulated by the combination attack. We see that all peaks disappeared in the transformed p-map, while the quality of the attacked image remains good. When we apply the combination attack to a dataset of 200 up-sampled images by a factor of 1.2, we found that the detection rate of [38] is reduced impressively to 3%.

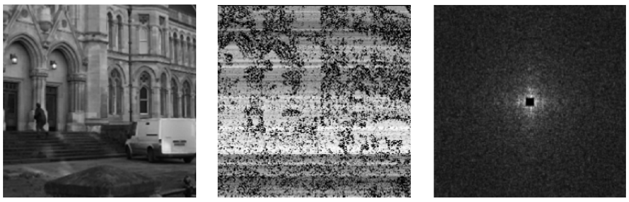


Figure 5-10: Detection results of the up-sampled image by a scaling factor of 1.2 and then post-processed by the combination attack.

5.4 A New Technique for Re-sampling Detection

5.4.1 Fast Re-sampling Detection

The core part of [38] is the EM algorithm used to estimate the probability of linear dependencies between neighboring samples. The results for all samples in the analyzed image are used to create the p-map. The remarkable peaks in the Fourier transform of the p-map become evidence to uncover traces of re-sampling and can be recognized easily by a peak detector.

Kirchner [102] showed that it does not matter what prediction weights (α) are used in the analysis, the linear prediction errors which determine the p-map will always be periodic in case of a re-sampled image. Thus, the author showed that the rather complex and time consuming EM estimation is not compulsory. As a result, he presented a fast but still reliable re-sampling detector that uses a pre-defined set of weights α. Kirchner [102] empirically found one of the best preset filter coefficients α for computation of the prediction error as:

$$\alpha = \begin{bmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{bmatrix}. \qquad \text{(Equation 5-1)}$$

Although the values of prediction weights do not affect the periodicity of the p-map, different sets of weights create different peak intensities in the p-map. For this reason, we call a p-map computed based on some predefined weights a pseudo p-map (pp-map for short). Through experiments, we found many times that using one predefined set of α in the algorithm of [102], peaks can be recognized in the transformed pp-map, but using another set, peaks are not evident (though the periodicity exists in theory). Consequently, the selected set of α strongly affects the obtained outcomes. Whilst the major advantage of [102] versus [38] is bypassing the EM estimation, we realize that the technique [38], where the intensities of the p-map are correctly computed is more robust and reliable.

5.4.2 The Proposed Technique

In this section, we introduce a re-sampling detection technique which consists of three main steps: 1) computing the pseudo probability map (pp-map), 2) applying the Radon transform to the pp-map and 3) detecting strong peaks in the Fourier domain of the pp-map. The detailed steps of the detection process are presented in Algorithm 5-1.

Probability Map Computation: The residue of a sample is computed following Equation 5-2 where the prediction weights are predefined in Equation 5-1. The probability of a pixel being correlated in the neighboring region is estimated based on the residue, modelled as a zero-mean Gaussian noise described in Equation 5-3. These steps compute the pp-map without using the EM algorithm as in [38].

Radon Transformation: Mahdian and Saic [41] improved the technique of Gallagher [40] by applying the Radon transform to the second derivatives of testing images. Accordingly, the technique [41] can detect not only rescaled images but also rotated images. The major drawback of [41] is its high false positive rate, especially when applied to images which containing strong textures. Inspired by the work of [41], we apply the Radon transform (see Section 4.4.1) to the pp-map of the image. To this end, firstly, the Radon transformation of the pp-map is computed for a set of predefined angles; this results in a set of projected vectors which are arranged in a matrix. If the image has been re-sampled, the corresponding auto-covariance matrix of the vectors exhibits a specific periodicity. Since our goal is to determine if an image has been subject to geometric transformations, we focus on the strongest periodic patterns present in the Fourier transform of the auto-covariance of the projected vectors. We assume that this technique works well for re-sampling detection due to the periodicity of the pp-map of re-sampled images shown in [102].

Peak Detection: After applying the Radon transform to the pp-map, we obtain a spectrum where critical peaks can easily be recognized. As an example, Figure 5-11 shows the results of applying the detector to an original image and a re-sampled image respectively. In order to infer the detection results, we search for strong peaks by computing the local maximums of the spectrum and infer the positions of the peaks based on a predefined threshold.

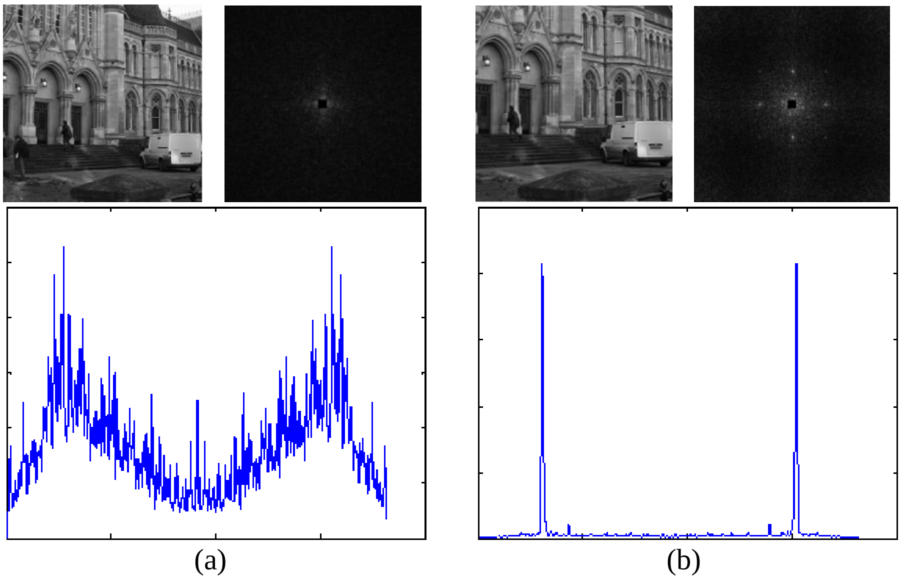


Figure 5-11: (a) Detection results of an original image, where the peaks in the Fourier spectrum are not clear; (b) detection results of the up-sampled image by a scaling factor of 1.2, where clear and strong peaks can easily be recognized.

Algorithm 5-1: Improved technique for re-sampling detection.

```
1. Input an image $y$ of size $M \times N$.
2. Initialize the parameters:
     ▪ Choose $\sigma$, $\theta$
     ▪ Set $\alpha$ = [-0.25, 0.5, -0.25, 0.5, 0.5, -0.25, 0.5, -0.25]
     ▪ Set $p_0 = \frac{1}{\max(y) - \min(y)}$
     ▪ Compute the pseudo probability matrix (pp-map).
   For each sample $i$
       $r(i) = \left| y(i) - \sum_{k=1}^{8} \alpha(k) y(i+k) \right|$          (Equation 5-2)
           where: $y(i + k)$ with k = 1, 2,…, 8 denote 8 neighboring samples of $y(i)$.
       $p(i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left( \frac{-r(i)^2}{2\sigma^2} \right)$          (Equation 5-3)
       $w(i) = \frac{p(i)}{p(i) + p_0}$
   End
3. Apply Radon transform following a set of angles $\theta$ to the pp-map $w$; this result in a
   matrix $R$, each column of $R$ is a vector $R_p$ of the Radon transform for one of the set of
   angles $\theta$.
4. Identify the evidence of re-sampling by locating the strongest periodic patterns present
   in the Fourier transformation of the auto-covariance of every $R_p$.
```

5.4.3 Experimental Results

Using the framework of Chapter 3, we tested the technique of [38] and our improved version with different image datasets of original images, re-sampled images and attacked re-sampled images. Firstly, we randomly collected 200 uncompressed images from [82], converted them to gray-scale and cropped each of them to 256×256 pixels in order to create a dataset of original images. From the dataset of original images, we created different datasets of up-sampled, down-sampled, and rotated images by different factors.

For our test, we use the set of weights (α) as in Equation 5-1. This set is also used as the initial weights in [38]. In both techniques, the size of the neighborhood is set to 3. In order to allow a fair comparison, we set the thresholds so that their detection rates in detecting up-sampled images by a factor of 1.2 are larger than 80% and their false positive rates when detecting original images are lower than 5%.

As presented in Section 5.3, the median filter is a strong attack against re-sampling detectors based on linear dependencies between neighboring samples. However, the major disadvantage of this attack is blurring. Among our targeted attacks, the hybrid median filter and the multiple re-sampling attack affects image perception quality less, but they seem not strong enough. The combination attack is more powerful, while still maintaining the image quality. The detection rates can be seen in Table 5-3. Both techniques work well to detect traces of re-sampling (with the detection rates of 99% and 83.5%) respectively and false positive rates below 5%. However, while the technique of [38] is mostly defeated by the combination attack with a detection rate down to 3%, our proposed technique is much more robust, as the detection rate remains over 50%. Consequently, in this section, we use only the combination attack in order to evaluate the security of the re-sampling detection techniques.

	No Attack	Median filtering	Hybrid median filtering	Multiple resampling	Combination attack
Popescu and Farid [38]	99.0%	1%	76.0%	84%	3.0%
Proposed	83.5%	25%	68.5%	66%	54.5%

Table 5-3: Detection rates when applying different attacks to up-sampled images by a scaling factor of 120%.

Next, we test both techniques with down-sampled images by different scaling factors. We realized that the detection rates of both techniques in detecting down-sampled images are rather low (Figure 5-12). The reason is that the down-sampling causes loss of information, thereby limiting the detection capacity.

We then evaluate the techniques with up-sampled images and rotated images as well as their attacked versions. The attacked images are created by applying the combination attack to the re-sampled images. We found that both techniques can detect up-sampled images by a scaling factor larger than 5% rather well (see Figure 5-13). The technique of [38] even detects up-sampled images by a factor larger than 10% perfectly (with a detection rate of nearly 100%). However, on the attacked images the detection rate of [38] is decreased significantly. This shows that [38] is not robust against this targeted attack. Although the proposed technique is not as powerful as [38] in detecting re-sampled images, it seems more robust against the combination attack. A similar situation occurs when rotated images are analyzed: both techniques work quite well once images are rotated by an angle larger than 3° (Figure 5-14). Although the proposed technique is little more robust than [38], both of them are almost defeated by the combination attack.

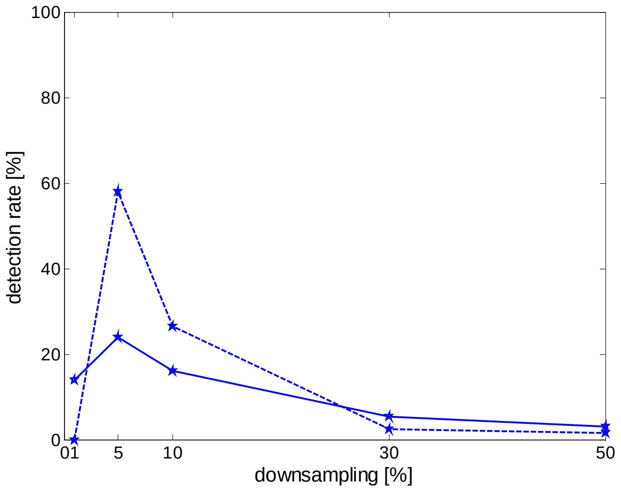


Figure 5-12: Detection rates for down-sampled images of our proposed technique (solid line), and the approach of Popescu and Farid [38] (dashed line).

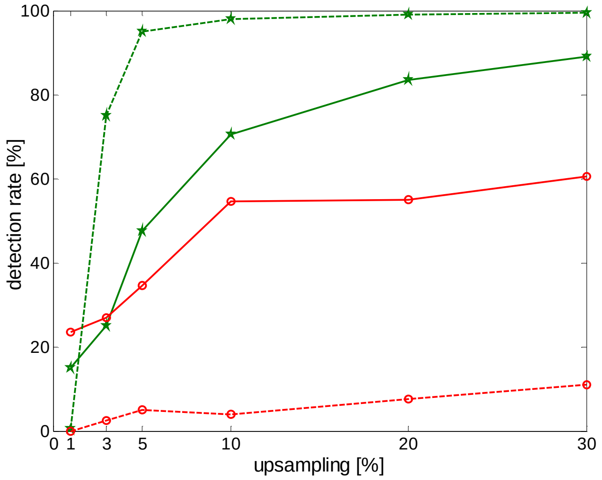


Figure 5-13: Detection rates for up-sampled images (dashed-star line for [38], solid-star line for our proposed technique) and for attacked up-sampled images (dashed-circle line for [38], solid-circle line for our proposed technique).

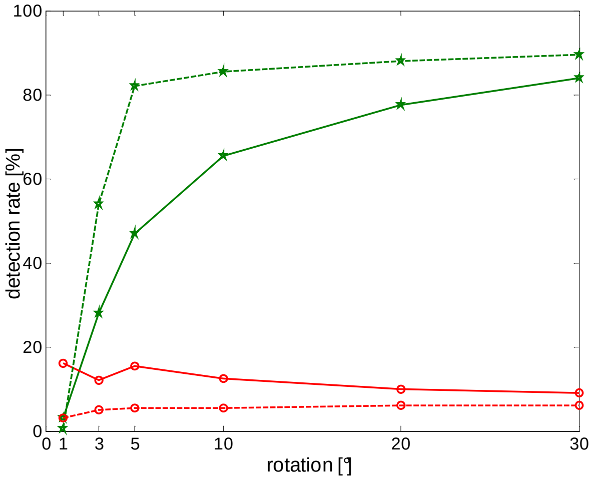


Figure 5-14: Detection rates for rotated images (dashed-star line for [38], solid-star line for our proposed technique) and for attacked rotated images (dashed-circle line for [38], solid-circle line for our proposed technique).

Due to using the Radon transform, our proposed technique is less sensitive to noise. To confirm this, we test the techniques with up-sampled images by a factor of 1.2 without any post-processing operation and with Gaussian noise addition. The results are shown in Table 5-4. While the detection rate of [38] is 99% in tests with up-sampled images, it is totally defeated when the images are post-processed by adding Gaussian noise by the SNR of 20 dB.

	No Attack	20 dB	25 dB	30 dB	35 dB
Popescu and Farid [38]	99.0%	1.0%	10%	36%	62.5%
Proposed	83.5%	36.5%	68%	77%	79.0%

Table 5-4: Detection rates for up-sampled images by a factor of 20% for and added Gaussian noise with SNR of 20 dB, 25 dB, 30 dB, and 35 dB.

	Add Noise SNR=25 dB	Median Filtering	Combination Attack
PSNR	21.20 dB	20.29 dB	22.93 dB
WPSNR	34.30 dB	32.74 dB	36.13 dB

Table 5-5: Difference between re-sampled images and attacked re-sampled images.

In order to quantify this aspect of an attack, we compute the average difference between pairs of re-sampled images (before the attack) and attacked re-sampled images (after the attack). The difference is measured by the PSNR and the WPSNR. A higher PSNR or WPSNR usually indicates that the attacked image is of higher quality. In Table 5-5, we show the average PSNR and WPSNR of 200 pairs of up-sampled images (by a factor of 1.2) and their versions under different attacks of adding Gaussian noise (25 dB), median filtering and the combination attack. We found that the combination attack maintains the best image quality among the test cases.

5.5 Summary

Re-sampling is involved in many image processing operations. Thus, detecting traces of re-sampling is an important problem in the field of image forensics. In this chapter, we discussed several re-sampling detection techniques [38], [40], [41] and used the test framework of Chapter 3 to evaluate the selected techniques. We designed some targeted attacks against the techniques in order to assess their security. We found that all analyzed techniques can be defeated by the combination attack. Lastly, we proposed a new re-sampling detection technique which offers better security in comparison with a state of the art technique of Popescu and Farid [38].

6 Forgery Detection in JPEG Images

6.1 Introduction

JPEG was proposed by the Joint Photographic Experts Group as a compression standard for continuous-tone still images, both gray-scale and color. The standard defines how an image is compressed into a stream of bytes and decompressed into an image for display. The JPEG standard is used in a number of image file formats such as EXIF (Exchangeable Image File Format) or JFIF (JPEG File Interchange Format). JPEG is mostly used in the form of lossy compression, where the compression rate can be adjusted. The JPEG file format is popular because of its efficiency of storage. In fact, most cameras in the market can export photos in JPEG file format and most tools for image editing support it.

Due to the popularity of the JPEG format, forgery detection for JPEG images became essential in digital image forensics. Although there are many ways of making forgeries in a JPEG image, most share three main steps: 1) loading the JPEG image which is compressed by a quality factor QF_1 to a photo editing software, 2) manipulating this image and 3) re-compressing it as a JPEG file with quality factor QF_2. Consequently, the re-saved image has been doubly JPEG compressed. Note that double JPEG compression (called D-JPEG) is not always a signal of malicious tampering: sometimes an image is re-compressed as JPEG with lower quality in order to save storage space or the image is re-saved after legitimate modification. Nevertheless, detection of D-JPEG can provide an important tool for authentication of JPEG images before further analysis [13].

Some authors [13], [42] found that when QF_1 is different from QF_2, periodic artifacts are present in the histograms of the DCT coefficients of D-JPEG images. This periodicity can be recognized in the Fourier transform domain through peaks in the spectrum. Based on this specific property, Popescu [13] proposed a technique to detect D-JPEG images. Lin et al. [110] expanded the global approach of [13] by locating tampered regions in the images. Bianchi et al. [111] developed an enhanced version of [110], leading to an improvement of the accuracy of the algorithm. Ye et al. [112] proposed a technique to estimate the local JPEG compression blocking artifacts by finding inconsistencies in the blocking artifacts. Some authors [113], [114] showed that the distribution of the most significant digit of the DCT coefficients in JPEG images follows the generalized Benford distribution. The distribution is very sensitive to the double JPEG compression and consequently this property can be applied to detect D-JPEG images. Farid [115] designed a technique to detect if a part of a JPEG image was initially compressed at a lower quality than the rest of the image. Chen et al. [116] proposed a set of image features, which have subsequently been evaluated by a SVM based classifier.

The aforementioned techniques are based on the characteristics of JPEG blocking artifacts; therefore, they can be attacked by destroying these artifacts. A popular attack is cropping: forensic techniques are usually defeated if the JPEG images are cropped before recompressing. The reason is that the corresponding blocking grids of the first compression and in the second compression are no longer aligned. To overcome this limitation, some more robust techniques have been proposed [44–46]. In [44] a blocking artifact characteristic matrix (BACM) is computed to measure the symmetric representation of the blocking artifacts introduced by JPEG compression. Since the symmetry of the BACM of a JPEG image is destroyed after the image is cropped, this artifact can be used as evidence for detecting cropped double JPEG compressed images. The authors in [45] model the linear dependency of the within-block pixels (the pixels that do not lie on the border of 8×8 blocks), compute the probability of the pixel being linearly correlated to its neighboring ones and form the map of the probabilities of all pixels in the image. They convert the map to Fourier domain, extract several statistical features from the different peak energy distribution and use the features to discriminate cropped D-JPEG images from non-cropped D-JPEG images. A simple yet reliable technique to detect the presence of cropped double JPEG compression has been introduced in [46]. This technique is based on the observation that the DCT coefficients exhibit an integer periodicity if the DCT is computed according to the grids of the primary compression. Through experiments, the authors showed that [46] outperforms other existing detection techniques for cropped D-JPEG images.

Although [44–46] work well for detecting cropped D-JPEG images, they will be defeated if the JPEG images are resized before the second compression step. Due to the effect of re-sampling, the blocking artifacts will be broken. The authors of [117] demonstrated the influence of resizing on the detection results of [114], [116]. There are a few techniques for detecting resized double JPEG compressed (RD-JPEG) images, among them [103], [118]. The authors of [103] apply a re-sampling detection technique (which was originally designed for use on uncompressed images) to JPEG images and analyze how the JPEG compression affects the detection output. A limitation of [103] is that its detection rates are very low when the JPEG quality factor of the first compression is larger than the quality factor used in the second compression. In addition, if the images are down-sampled before re-compression, the technique is mostly defeated. The technique of [118] extracts neighboring joint density features and applies Support Vector Machines for detecting RD-JPEG images. Although the technique works for both up-sampled images and down-sampled images, it is analyzed only by the quality factor of 75 and no information on the false positives is given. Bianchi and Piva [119] proposed an algorithm, which can be summarized by some steps: 1) estimate the candidate resizing factor; 2) for each candidate factor, undo the image resizing operation and measure the NLDP (near lattice distribution property); 3) if the result is greater than a predefined threshold, label the image as resized double JPEG compressed. Furthermore, the technique [119] can estimate both the resize factor and the quality factor of the first JPEG compression of the analyzed image. The experimental results in [119] show that it surpasses [103] on the same test condition, but similar to [103], it seems more difficult to detect when QF_1 is much larger than QF_2.

In this chapter, first we introduce the problem for detection of double JPEG compression and then we apply the evaluation framework in Chapter 3 to assess the effectiveness and security of selected forensic techniques for JPEG images: one for doubly JPEG compressed images [13] and one for cropped doubly JPEG compressed images [46]. In order to measure the effectiveness and security of the techniques, we design different targeted attacks. We show that the evaluated techniques work quite well in the case no attack is applied to the forged image, but they are vulnerable and can easily be defeated by simple attacks. Lastly, we propose an improved technique to detect RD-JPEG images. The technique reveals specific features of JPEG images and RD-JPEG images by using a re-sampling detector. The extracted features are fed to SVM-based classifiers in order to discriminate RD-JPEG images from JPEG images.

6.2 Double JPEG Compression

6.2.1 DCT-based Compression

In order to compress a color image (RGB) in the JPEG format, firstly the image is transformed from RGB into a luminance-chrominance color space (YCbCr), which consists of one luminance channel (Y) and two chrominance channels (Cb, Cr). The chrominance channels are typically down-sampled by a factor of two and each channel is independently processed. The process of JPEG compression consists of three main steps of applying the Discrete Cosine Transform (DCT), quantization and entropy coding (Figure 6-1) [120]. The JPEG compression of a single channel image (e.g. gray-scale image) is similar to the process of one channel of a color image.

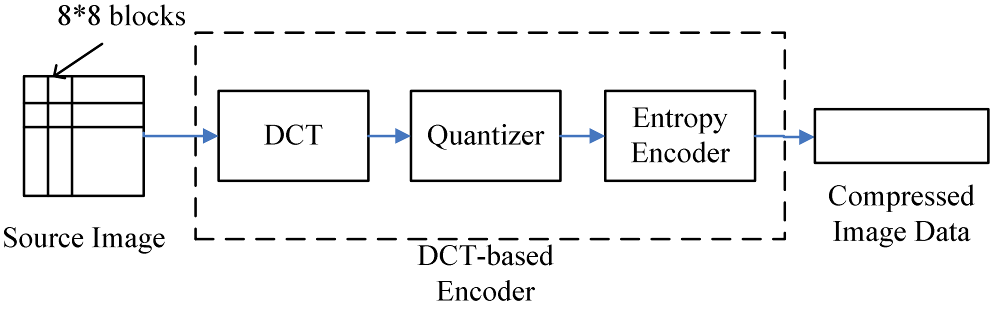


Figure 6-1: JPEG encoding steps.

Discrete Cosine Transform (DCT): The image is divided into 8×8 non-overlapping blocks and shifted from unsigned integers with range [0, 255] to signed integers with range [-128, 127]. Finally, the DCT is applied to every block. Let $f(x,y)$ denote an 8×8 image block, the two-dimensional Discrete Cosine Transform (DCT) of the block is computed as follows:

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16} \ ,$$

where $u, v \in \{0,1,\ldots,7\}$,

$C(u), C(v) = \frac{1}{\sqrt{2}}$ for $u, v = 0$ and

$C(u), C(v) = 1$ otherwise.

Quantization: Each coefficient in a DCT block is divided by a quantization factor specified in of a quantization table and rounded to the nearest integer. The purpose of this step is to achieve compression by discarding information which is not visually significant. Since quantization is a non-invertible operation, this step causes loss of information each time an image is JPEG compressed. The quantized coefficient at (u, v) is computed as follows:

$$F_1(u,v) = \left[\frac{F(u,v)}{q(u,v)} \right] ,$$

where $[X]$ denotes rounding of X to the nearest integer and $q(u, v)$ is a quantization factor. The quantization factor depends on the JPEG quality factor and can be computed based on the standard quantization table (see Appendix A1).

Entropy coding: This step achieves additional lossless compression by encoding the quantized DCT coefficients in a compact form based on their statistical characteristics. The JPEG proposal specifies two entropy coding methods namely Huffman coding and arithmetic coding. This step is reversible and therefore does not affect the image quality.

Decoding a JPEG image consists of performing similar steps of encoding, but in reverse order: entropy decoding, de-quantization and applying the Inverse Discrete Cosine Transform (IDCT) (Figure 6-2).

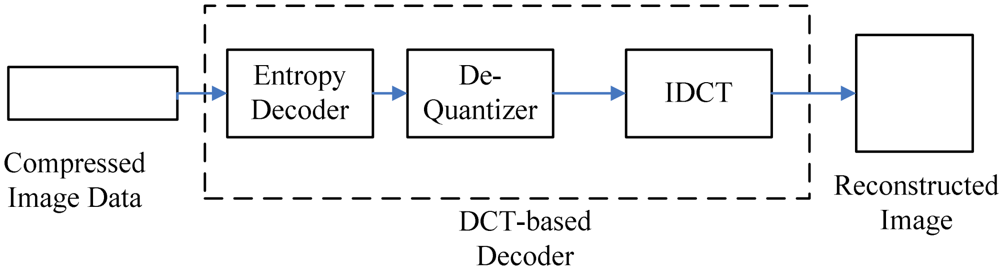


Figure 6-2: DCT-based decoding steps.

Entropy decoding: The compressed data is decoded in order to recover the quantized DCT coefficients.

De-quantization: The quantized coefficients are multiplied by the corresponding quantization factors.

Inverse Discrete Cosine Transform (IDCT): The IDCT of the de-quantized DCT coefficients are computed to obtain the reconstructed image block $f'(x,y)$:

$$f'(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F'(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \ ,$$

where $u, v \in \{0,1,\ldots,7\}$,

$C(u), C(v) = \frac{1}{\sqrt{2}}$ for u, v = 0 and

$C(u), C(v) = 1$ otherwise.

Finally, the result $f'(x,y)$ is shifted back to the range [0, 255] in order to reconstruct an 8×8 image block.

6.2.2 Double Quantization

In the JPEG compression process, the quantization step is non-invertible and introduces specific artifacts that can be used in forensics. When the decoded image is compressed one more time, a similar encoding and decoding process is applied to the image. As a result, in a double JPEG compressed image, the DCT coefficients can be expressed as follows:

- After the first compression, the DCT coefficient $F(u, v)$ is divided by a quantization factor $q_1(u, v)$ and rounded to the nearest integer:

$$F_1(u,v)=\left[\frac{F(u,v)}{q_1(u,v)}\right].$$

- After the first de-compression, the quantized coefficient is multiplied by the same quantization step of $q_1(u, v)$, resulting in a value $F'(u,v)$ that is slightly different from $F_1(u, v)$:

$$F'(u,v)=\left[\frac{F(u,v)}{q_1(u,v)}\right]q_1(u,v).$$

- After the second compression, the image is DCT transformed and the DCT coefficients are quantized by another quantization factor $q_2(u, v)$:

$$F_2(u,v)=\left[\left[\frac{F'(u,v)}{q_1(u,v)}\right]\frac{q_1(u,v)}{q_2(u,v)}\right].$$

Popescu [13] showed that the histogram of certain DCT coefficients in a double JPEG compressed image is periodic and the periodicity is visibly in the Fourier transform. The author proposed a technique uses these specific characteristics in order to determine whether an image has been double JPEG compressed. In the next section, we describe the technique and evaluate its effectiveness.

One can note that when an image is re-compressed with the same quality of the first compression, double quantization artifacts mostly do not appear. Thus, the technique of Popescu [13] does not work in this case. To overcome this problem, Huang et al. [121] proposed a technique for detecting D-JPEG images when the first and the second quality factor are the same. This technique is based on the observation that when re-compressing a JPEG image over and over again, the number of different quantized DCT coefficients between the sequential two versions will monotonically decrease in general even though the first and the second quality factors are the same. Subsequently, the authors find a randomly perturbation ratio, which can be used to discriminate single images from double JPEG compressed images.

6.2.3 Detection of Double JPEG Compression

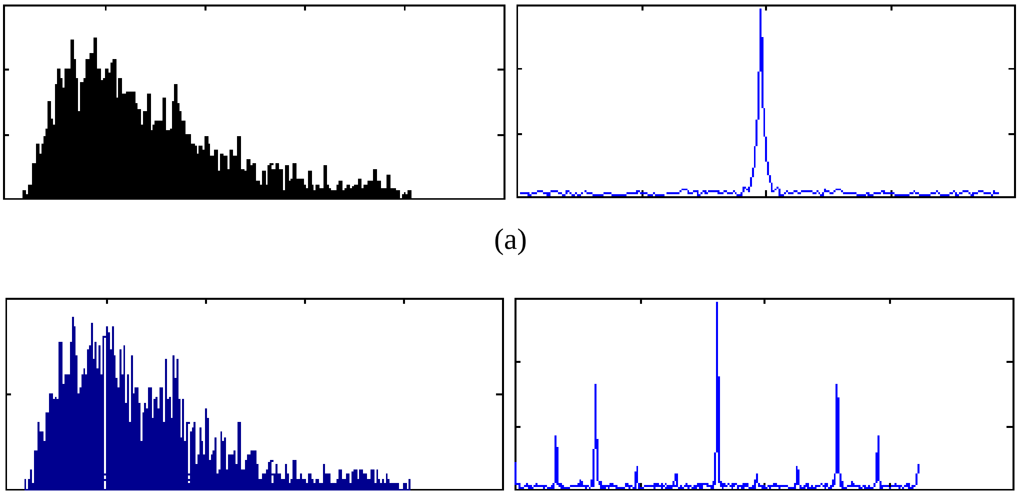
The technique of Popescu [13] is based on computing the histograms of low frequency DCT coefficients of the image and detecting periodic artifacts. Since this technique is widely-used for detecting double JPEG compressed images, we briefly review and evaluate the technique before showing that the technique is not robust against a simple attack (cropping). Through experiments, we found that artifacts are most evident in the histogram of DC coefficients. For simplicity, we take into account only the DC coefficients of DCT blocks in our experiments.

The algorithm [13] to detect whether an image is D-JPEG compressed can be summarized as follows:

1. Dividing the image into 8×8 non-overlapping blocks.
2. Applying the DCT to every block.
3. Taking the DC coefficient from every DCT block and computing the histogram of the coefficients.
4. Computing the Fourier transform the histogram; normalizing the histogram and looking for strong peaks in the spectrum as evidence for double JPEG compression.

Figure 6-3 shows the histogram of DCT coefficients of a sample image and its Fourier transform in the cases of single JPEG compression and double JPEG compression. Apparently, in the case a single compression, there are no strong peaks in the Fourier transform, but the peaks are obvious after a double compression took place.

In order to quantify the sensitivity and robustness of this algorithm, we use a simple method based on detection of strong peaks in the histogram of DC coefficients instead of using a complex method based on a parameterized Laplace model and a least square optimization as advocated in [13].



(a)

(b)

Figure 6-3: (a) Histogram of DC coefficients of a sample image and its Fourier transform (image compressed by a factor of 70); (b) Histogram of DC coefficients and its Fourier transform of an image which has been created by JPEG compressing the image (a) by the factor of 90.

6.2.4 Experimental Results

In this section, we evaluate the double JPEG compression detection algorithm of Popescu [13] using the general test framework presented in Chapter 3. Firstly, a dataset of original images was created by randomly choosing 200 never-compressed images from the UCID image database [82]; all selected images are converted to gray-scale. Subsequently, datasets of JPEG compressed images are formed by compressing the original images by different quality factors QF_1 ($QF_1 = 50, 55, \ldots, 90, 95$). Then each single JPEG compressed image is re-compressed in JPEG format using other quality factors QF_2 ($QF_2 = 50, 55, \ldots, 90, 95$) and $QF_1 \neq QF_2$. This step creates a set of suitable double JPEG compressed images.

The detection rates of the technique in detecting D-JPEG compressed images are shown in Table 6-1 (the false positive rates of the technique when testing on the dataset of single JPEG compressed images are lower than 20%). The technique works well when $QF_1 < QF_2$ and the difference between QF_1 and QF_2 is large. When $QF_1 > QF_2$ the detection rate of the technique is rather low and when $QF_1 = QF_2$, there are no specific artifacts to be detected.

	50	55	60	65	70	75	80	85	90	95
50	-	51.5	76.0	86.5	89.0	37.0	100	99.0	100	100
55	27.5	-	29.0	85.0	93.5	68.5	95.5	100	100	100
60	37.0	28.5	-	57.5	78.0	63.0	76.5	99.5	100	100
65	40.5	43.5	32.0	-	26.0	60.0	56.5	75.0	99.5	100
70	30.0	43.5	40.5	26.5	-	50.5	88.0	76.5	98.5	100
75	24.0	31.0	34.0	42.0	46.5	-	55.5	75.5	100	100
80	25.0	25.0	27.5	24.5	45.0	25.5	-	57.5	34.0	57.5
85	24.5	23.5	26.5	24.0	24.0	25.0	27.5	-	54.0	78.5
90	23.5	25.5	24.0	24.0	24.0	24.0	23.5	31.0	-	35.0
95	26.0	25.0	23.0	22.5	22.5	23.5	22.5	22.5	26.5	-

Table 6-1: Detection rates of [13] in detecting double JPEG compressed images (QF_1 in rows and QF_2 in columns) [%].

6.3 Security Analysis

6.3.1 Security of Double JPEG Compression Detection

In this section, we evaluate the security of the detection technique for D-JPEG compression [13]. To this end, firstly, we apply a targeted attack that aims at removing the specific artifacts in D-JPEG images.

Since the technique of [13] is based on the characteristics of JPEG blocking artifacts, an effective attack to the technique is cropping the image (usually by a few pixels to avoid being exposed) before the second compression. After cropping, the DCT blocks in the second compression are not aligned any more with the corresponding DCT blocks in the first compression. As a result, the specific artifacts of double compression are likely destroyed. The author of [13] mentioned this problem, however, but performed no experimental evaluation to confirm the fact.

To evaluate the security of the algorithm of [13], the cropping attack is applied while creating double JPEG compressed images. To this end, double JPEG compressed images (by the quality factors of 55, 70 and 85) are selected, and each of them is cropped by a random number of pixels. Finally, the images are re-compressed as JPEG by different quality factors in order to create non-aligned double JPEG compressed images. Our experimental results are shown in Figure 6-4, Figure 6-5 and Figure 6-6. The results indicate that the forensic technique works well when no attack is applied but it is not robust against cropping and is mostly defeated by this attack.

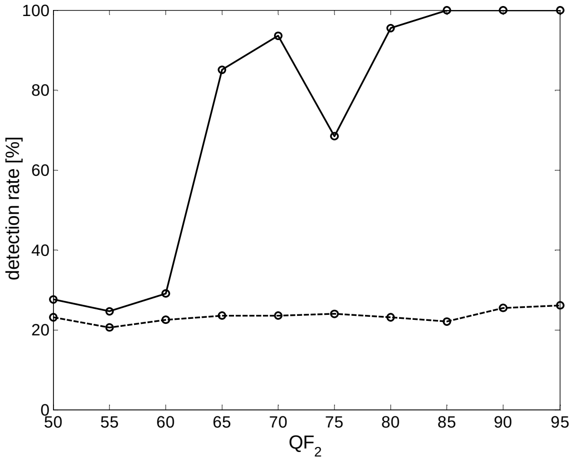


Figure 6-4: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 ($QF_1 = 55$).

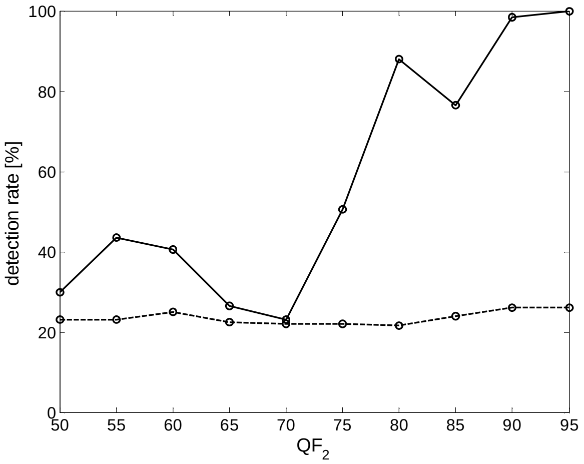


Figure 6-5: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 (QF_1 = 70).

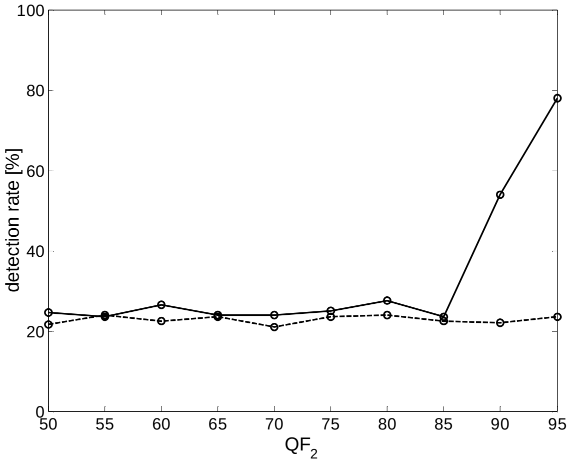


Figure 6-6: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 (QF_1 = 85).

6.3.2 Cropped Double JPEG Compression Detection

Although cropping is a simple and effective to attack to D-JPEG compression detection, this operation leaves other types of evidence, which can be used to detect whether an image has been cropped. Several authors have proposed techniques to detect cropped D-JPEG images, among them [44] [45][46]. Through experiments, Bianchi and Piva [46] showed that their technique outperforms other existing ones [44][45]. In this section we revisit the technique of Bianchi and Piva [46] and then apply the evaluation framework to assess its effectiveness and security.

The authors of [46] showed that, if an 8×8 block DCT is applied to the JPEG image, there are three possible cases: the grid is aligned with the last JPEG compression, the grid is aligned with the first JPEG compression and the grid is misaligned with the two previous ones. The authors showed that when the block DCT grid is aligned with the DCT grid of the last compression or the first compression, the coefficient histograms tend to be periodic. In theory, the effect described above can be measured for every DCT coefficient; however, through experiments the authors observed that it is more evident in the case of DC coefficient. Based on this observation, the authors proposed an algorithm to detect cropped D-JPEG images as follows:

1. Apply DCT to the test image with every possible shift (i, j) and $i, j = \{0, 1\ldots 7\}$ (when i=0 and j=0, the block DCT grid is aligned with the DCT grid of the last compression).
2. For each value of the shift (i, j), compute the histogram of DC coefficients of every block of the image. After that, the periodicity of the histogram is evaluated by applying the Fourier transform.
3. Compute the Integer Periodicity Map (IPM), which is formed by the proportion of the magnitude of the periodicity of the histogram in a particular shift to the sum of the magnitude of the periodicity of all possible shifts.
4. Compute the uniformity of each IPM by measuring its min-entropy. Min-entropy is a popular metric in statistics which characterizes the most probable occurrence of a random variable. It is easy to verify that a high min-entropy corresponds to a mostly uniform IPM, whereas an IPM with a high peak will be characterized by a low min-entropy.
5. The presence of cropped D-JPEG is detected by applying a threshold detector to the min-entropy of the IPM, measuring its uniformity. When the min-entropy is smaller than a suitable threshold, it is concluded as cropped D-JPEG.

6.3.3 Experimental Results

The technique [46] based on a specific artifact present in cropped D-JPEG images. Apparently, the artifact will be destroyed if the DCT coefficients of a cropped D-JPEG image have been altered. Several methods can be used for this purpose, such as geometric transformation or noise addition. In this work, we use rescaling because this operation affects visual quality less. In particular, we rescale cropped images before the second compression. It is well known that scaling is a popular attack which also used in some other situations [74], [117].

Similar to the previous tasks, we use the UCID dataset [82] for experimental analysis. Firstly, 200 images are chosen randomly from the UCID dataset, and converted to gray-scale. After that, JPEG compressed datasets are created by compressing the original images by different quality factors QF_1 (QF_1 = 50, 70, 85). Each single JPEG compressed image is cropped randomly by i pixels horizontally and j pixel vertically ($0 \le i, j \le 7$) and then re-compressed in JPEG format by another quality factor QF_2 (QF_2 = 50, 55, …, 90, 95) and $QF_2 \neq QF_1$. This creates a dataset of cropped D-JPEG images. An attacked image is created by cropping randomly a single JPEG compressed image, rescaling the image to the previous size and then re-compressing the image in JPEG format with a different quality factor QF_2 (QF_2 = 50, 55, …, 90, 95) and $QF_2 \neq QF_1$. The technique [46] is evaluated on the cropped D-JPEG images and the attacked cropped D-JPEG images. The experimental results are shown in Figure 6-7, Figure 6-8 and Figure 6-9. The technique works well when $QF_2 > QF_1$ and the difference between QF_1 and QF_2 is large. However, it is mostly defeated by the attack.

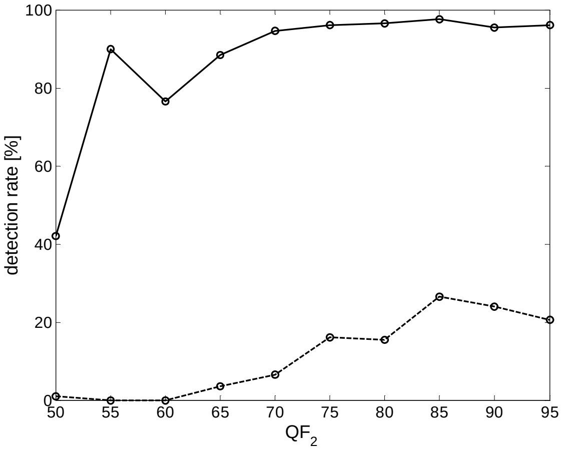


Figure 6-7: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 55$.

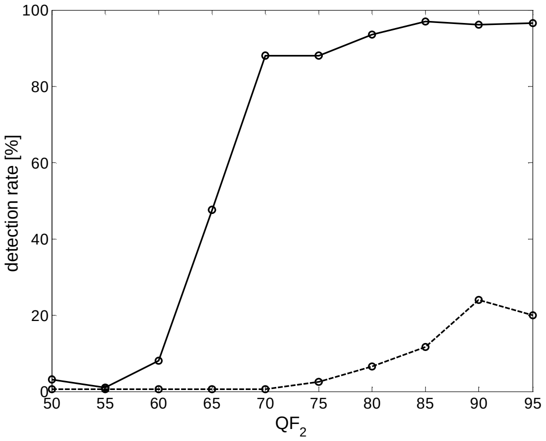


Figure 6-8: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 70$.

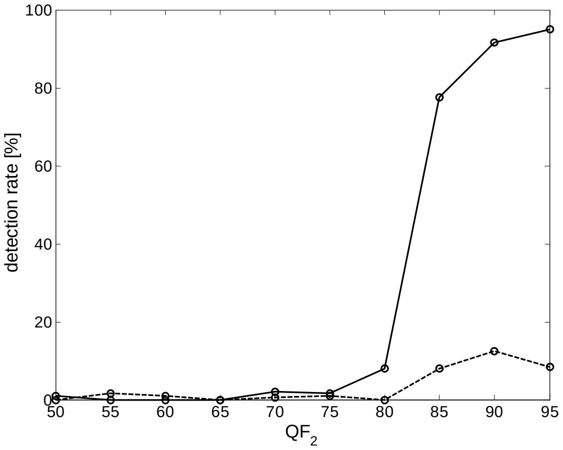


Figure 6-9: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 85$.

6.4 A New Technique for Resized Double JPEG Compression Detection

6.4.1 The Proposed Technique

When using [41] to detect re-sampling in both JPEG images and RD-JPEG images, we empirically found that in the detection result of RD-JPEG images seems to have more peaks than that of JPEG images (Figure 6-10). This is because the detection result of a RD-JPEG image contains not only the peaks introduced by JPEG compression, but also the peaks due to re-sampling. Nevertheless, the difference is not always easy to recognize by human eyes. In addition, it is necessary to automatically classify RD-JPEG images from JPEG images. To this end, we first apply the technique [41] to JPEG images, and then extract the values of maximal peaks from the normalized Fourier spectrum. The extracted features are subsequently fed to SVM-based classifiers in order to discriminate RD-JPEG images from JPEG images. Since SVM is only a binary classifier, we use two approaches to design SVM classifiers.

In the first approach, we design a single SVM classifier for directly distinguishing JPEG and RD-JPEG images, compressed by different quality factors. To this end, the features of a set of JPEG images and their re-sampled versions (the number of JPEG and re-sampled JPEG images are the same) are extracted for training a SVM classifier. This approach is simple and suitable for many situations in practice when we do not know the quality factors of the analyzed images. However, through experiments, reported in Section 6.4.2, we find that this technique works well mostly when QF_1 is lower than the QF_2.

The second approach is based on the idea that while QF_1 of a double JPEG compressed image is usually not known to the analyst, QF_2 can be identified (we present a method to reveal the last quality factor of a JPEG image in Appendix A1). Thus, instead of using a single classifier for all, we design several different SVM classifiers which each distinguish JPEG and RD-JPEG images for one specific value of QF_2. Once the last quality factor of an analyzed JPEG image is known, the corresponding classifier will be applied to it. The method to design a classifier for a particular QF_2 is similar to the first approach: we first use a set of JPEG images and another set of RD-JPEG images (the numbers of images in both sets are the same and every image is compressed by QF_2) and then extract image features for training a SVM classifier. In other words, the last quality factor of a tested image is first identified, and then the image will be analyzed by the corresponding qualifier.

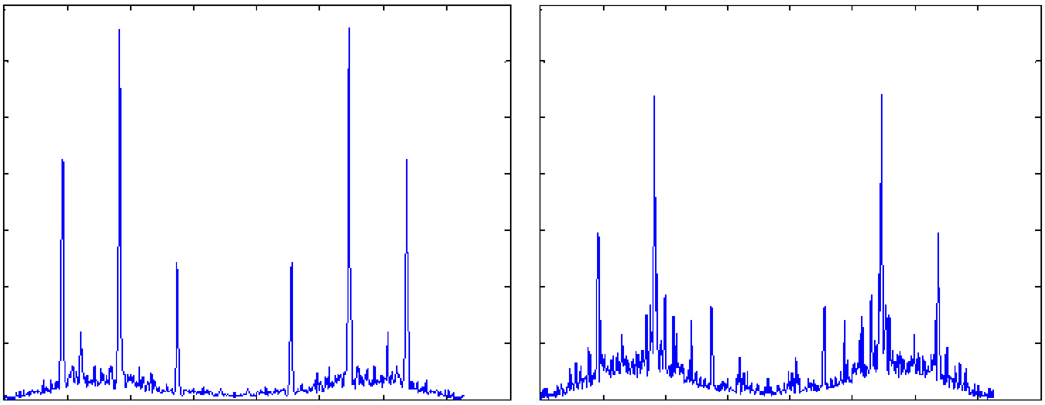


Figure 6-10: Shown on the left is the detection result of the JPEG image of Lena and on the right the detection result of the RD-JPEG version of the same image.

6.4.2 Experimental Results

First, we randomly choose 200 uncompressed images from the UCID image database [82]. We create 5 datasets of JPEG images by compressing the uncompressed images with the quality factors of 40, 50, 60, 70, and 80. The JPEG images are subsequently resized by a scaling factor of 1.2 and recompressed by different factors of 40, 50, 60, 70, and 80. As a result, we obtained 5 datasets of RD-JPEG images corresponding to each dataset of JPEG images.

To test the first approach, we create a single SVM classifier by using two groups of JPEG images and RD-JPEG images (with the scaling factor of 1.2) for training. After the training process we apply the classifier to test RD-JPEG images. In training, we consider two cases of different quality factors: 1) 100 JPEG images compressed by a quality factor of 50 and 100 RD-JPEG images re-compressed by a quality factor of 70 ($QF_1 = 50$, $QF_2 = 70$ and scaling factor = 1.2) and 2) 100 JPEG images compressed by a quality factor of 70 and 100 RD-JPEG images re-compressed by a quality factor of 80 ($QF_1 = 70$, $QF_2 = 80$ and scaling factor = 1.2). Analyzing the detection results (Table 6-2 and Table 6-3), we found that the technique works well when detecting RD-JPEG images where QF_1 is smaller than QF_2. Otherwise, when QF_1 is larger than QF_2, the detection rate is small. In our experiments, the false positive rates (computed by testing the classifier on datasets of JPEG images which have been compressed by different quality factors of 40, 50, 60, 70, and 80) are lower than 11% in the first case and lower than 8% in the second case.

In a more realistic scenario, we test the techniques on the RD-JPEG images, which have been resized with a different scaling factor than the factors are used in the training process. The datasets are created in the same way as above, except the scaling factor 1.1 is used instead of 1.2 (i.e. QF_1=70, QF_2=80 and scaling factor =1.1). Although the detection results (in Table 6-4) are clearly worse compare with Table 6-2 and Table 6-3, we found that the degradation is not significant; therefore, the technique can potentially work in case the scaling factor is unknown.

In the second approach, we consider 5 different cases corresponding to a QF_2 of 40, 50, 60, 70, and 80. When QF_2 is 40, we organize the training images into two groups: a group of 100 JPEG images (the quality factor of 40) and the other group of 100 RD-JPEG images (QF_1= 50, QF_2 = 40 and scaling factor = 1.2). The extracted features are used to train a SVM classifier that can be used to detect RD-JPEG images compressed by the QF_2 of 40. We repeat this process for the other cases when QF_2 is 50, 60, 70, and 80. The detection results in testing RD-JPEG datasets are presented in Table 6-5. We noticed that following the second approach, the technique works well even if QF_1 is larger than QF_2. For example, when QF_1 = 80 and QF_2 = 40, in the first approach, the detection results are only 10.5% or 24%, but in the second approach, it reaches 85.0%. The false positive rates are lower than 10% (9%, 8%, 5%, 6% and 3% when testing JPEG images compressed by the quality factors of 40, 50, 60, 70, and 80 respectively).

	40	50	60	70	80
40	65.5%	91.0%	99.5%	99.5%	84.5%
50	52.5%	80.0%	97.0%	99.0%	87.0%
60	35.5%	77.5%	92.5%	98.5%	88.0%
70	19.5%	67.5%	87.0%	99.0%	84.0%
80	10.5%	45.0%	79.5%	91.5%	78.0%

Table 6-2: Detection results using a single SVM classifier (training JPEG images compressed by QF = 50 and RD-JPEG images re-compressed by QF_1 = 50, QF_2 = 70) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	70.0%	94.0%	98.5%	99.0%	95.0%
50	62.0%	80.0%	92.5%	98.5%	98.0%
60	48.0%	76.0%	87.5%	96.5%	99.0%
70	33.5%	68.0%	83.0%	93.5%	99.0%
80	24.0%	57.0%	69.0%	81.0%	92.0%

Table 6-3: Detection results using a single SVM classifier (training JPEG images compressed by $QF = 70$ and RD-JPEG images re-compressed by $QF_1 = 70$, $QF_2 = 80$) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	37.0%	57.0%	63.5%	78.0%	82.5%
50	37.0%	58.0%	63.5%	78.5%	83.0%
60	26.0%	48.0%	66.5%	77.5%	87.0%
70	13.5%	43.5%	68.0%	77.5%	73.0%
80	10.5%	39.0%	62.5%	77.0%	86.5%

Table 6-4: Detection results using a single SVM classifier (training JPEG images compressed by QF=70 and RD-JPEG images re-compressed by QF_1=70, QF_2=80) for RD-JPEG images by the scaling factor of 1.1 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	95.0%	91.5%	89.5%	99.0%	98.0%
50	90.0%	90.0%	88.5%	98.5%	99.5%
60	89.5%	91.0%	97.5%	98.0%	100%
70	87.5%	85.0%	95.0%	99.5%	98.0%
80	85.0%	80.0%	96.0%	100%	99.0%

Table 6-5: Detection results using dedicated SVM classifiers for RD-JPEG images (depending on the quality factor of the second compression) by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

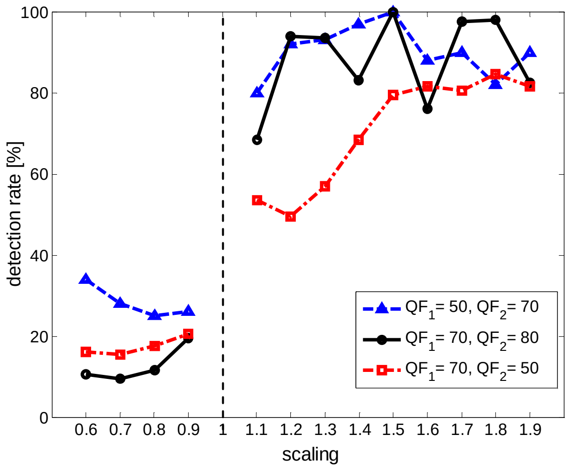


Figure 6-11: Detection results for RD-JPEG images by different scaling factors when the quality factor of the trained images and the test images are the same.

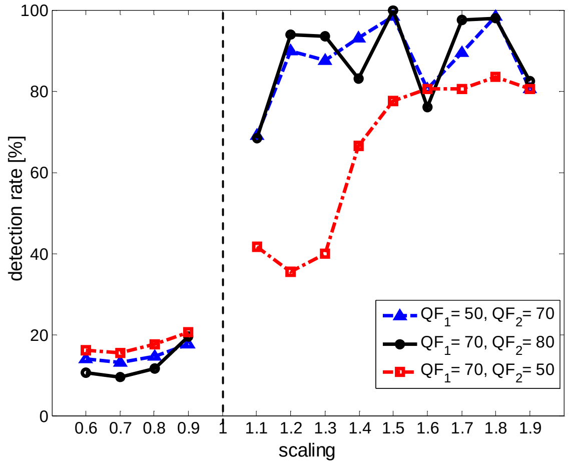


Figure 6-12: Detection results for RD-JPEG images by different scaling factors when the quality factor of the trained images and the test images are different.

In order to assess the influence of scaling factor, we test the proposed technique for detection of RD-JPEG images with various scaling factors. The RD-JPEG images are created by resizing JPEG images (firstly compressed by QF_1) of different scaling factors (from 0.6 to 1.9) and then they are recompressed (by a different quality factor QF_2). We consider three cases: 1) QF_1=50 and QF_2=70, 2) QF_1=70 and QF_2=80 and 3) QF_1=70 and QF_2=50. We create different datasets of JPEG images and RD-JPEG images and in each case, the training and testing processes of the classifiers are conducted as described before. The detection results in various scaling factors are shown in Figure 6-11. Due to missing information in the down-sampling process, the detection rates of the down-sampled images are very low. Detecting up-sampled images is possible with much higher rates. In some cases, the detection rates even reach about 100%. In this scenario, the test images are compressed with the same quality factors as the training images (but with different scaling factors). We found that scaling factors affect the detection results: typically the detection rates tend to increase.

Lastly, in a more realistic scenario, we apply the technique trained by one image type (QF_1=70, QF_2=80, scaling factor = 1.2) to images with different types (QF_1=50 and QF_2=70, QF_1=70 and QF_2=50, and scaling factor ranges from 0.6 to 1.9). The detection results are presented in Figure 6-12. Although the results deteriorate (compare with Figure 6-11), we found that the degradation is not significant; therefore, the technique can potentially work in a real condition.

6.5 Summary

In this chapter, we designed a technique for detecting resized double JPEG compressed images. The technique is based on applying the re-sampling detector [41] to JPEG images, and extracting features from strong peaks of the normalized Fourier transformation. Then the extracted features are fed into a SVM-based classifier in order to discriminate RD-JPEG images from JPEG images. We propose two methods to design SVM classifiers: one single global classifier and several classifiers depending on the quality factor of the last compression. Although the first approach is simple and easy to use, the second approach achieves higher detection rates. In comparison with [103], our technique has higher detection rates when the quality factor of the first compression is larger than the quality factor of the last compression and when detecting down-sampled double JPEG compressed images. We apply the technique to test RD-JPEG images resized with different scaling factors and found that the scaling factors can affect the detection results.

7 Conclusions and Future Work

In this thesis, we have addressed the security of digital image forensic algorithms, which is ability to withstand dedicated attacks that aim at making an attacked image look authentic. In order to measure the effectiveness and security of forensic techniques, we developed a test framework. The framework provides the necessary infrastructure and support tools, which allow evaluating forensic techniques in an automatic way. Since forensic techniques in the same category are tested in the same condition, their effectiveness and security can be fairly compared to each other.

We implemented several image forensic techniques based on published algorithms. For each technique we designed different targeted attacks and used them in the evaluation framework. Targeted attacks against a forensic technique allow analyzing the security of the forensic techniques and providing more insight into their use.

Once a forensic algorithm is publicly known, any forensic tool that is based on the algorithm can be attacked. Therefore, developing forensic techniques which offer a higher level security is an urgent need. In this thesis, we designed a number of new forensic techniques in different categories of forgery detection. Through experiments, we showed that our techniques are more robust against dedicated attacks in comparison with some state-of-the-art image forensic techniques. Obviously, adversaries can develop new attacks targeted our techniques and potentially disable them. Although forensics and anti-forensics seems to make a never-ending game, research on security of forensic techniques allows developing more reliable techniques.

The purpose of attacks is to remove or destroy evidence of forgeries in digital images, but the attack itself can leave specific artifacts. Thus, detecting such artifacts can reveal the presence of the attacks. Little work has been done in this direction up to now [60], [65], [122]. Along with the development of anti-forensic methods, in order to thoroughly understand the security of forensic techniques, countering anti-forensics should be investigated. This is one of our future works. In addition, we will improve some of the algorithms presented in this work. We will not only focus on individual attacks but also consider more about the combination of different attacks. We think carefully about forensics as a classification problem, which can be solved by using different techniques, e.g. machine learning.

Appendix

A1. Determining the Last Quality Factor of a JPEG Image

The compression ratios of JPEG images are controlled by the quantization tables which used in the compression process. In this thesis, we focus on images stored in the JPEG Interchange File Format (JFIF). The JFIF is the most commonly used format for JPEG data [123]. The quantization table that was used to compress an image is stored in the JFIF header [124]. This table (called *Ts*) can be identified by using the JPEG Toolbox [125].

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Table 8-1: The standard quantization table

The most commonly used standard quantization tables are published by the International JPEG Group (IJG). Based on the standard table (*Tb*), shown in Table 8-1, and the quality factor (*Q*), the quantization table can be computed as follows:

$$S = \begin{cases} \dfrac{500}{Q} & \text{if } Q < 50 \\ 200 - 2Q & \text{otherwise} \end{cases}$$

$$Ts[i] = \left\lfloor \frac{S * Tb(i) + 50}{100} \right\rfloor .$$

Conversely, when the tables *Tb* and *Ts* are known, the approximate value of the quality factor can be computed as follows [124]:

$$S' = \frac{Ts(i) * 100 - 50}{Tb(i)} ,$$

$$Q' = \begin{cases} \left\lfloor \dfrac{200 - S'}{2} \right\rfloor & \text{if } S' \le 100 \\ \left\lfloor \dfrac{5000}{S'} \right\rfloor & \text{otherwise}. \end{cases}$$

Note that the function to predict the quality factor involves integer computation on the quantization table (*Ts*) that introduces integer rounding errors, so the value of *Q'* is closely to *Q*. Following a suggestion in [124], then the computed quality factor (*Q'*) should be off by one or two.

For example, if we know the quality factor $Q = 80$ and $Tb(1) = 16$, then $S = 40$, therefore, $Ts(1) = 6$. Conversely, if we know $Tb(1) = 16$ and $Ts(1) = 6$, then $S' = 34.375$, therefore, $Q' = 82$.

List of Figures

Figure 1-1: Shown on the left is the forged image of the US President Abraham Lincoln, which is a composite of the head of President Lincoln and a picture of the body of the southern politician John Calhoun (on the right). 11

Figure 2-1: Watermark embedding process. 15

Figure 2-2: Watermark detection process. 16

Figure 2-3: (a) Circa 1865: shown on the left is the forged photograph after General Francis P. Blair was added at the rightmost position and shown on right is the original photograph; (b) 1936: shown on the left is the forged photograph after removing Po Ku and shown on the right is the original photograph.......... 19

Figure 2-4: (a) 2007: the retouched photograph of Roddick in *Men's Fitness* Magazine; (b) 2007: a copy-move forged picture on the cover of *Nature*; (c) 2007: manipulated result of the group of professor Michael Roberts in a paper published in the journal of *Science*. . 20

Figure 2-5: 2011: banned advertisements by the ASA of (a) Christy Turlington, (b) Julia Roberts and (c) Natalie Portmann.......... 20

Figure 2-6: The pipeline for image capturing in digital cameras.......... 21

Figure 2-7: A computer generated picture of Song Hye Kyo. 22

Figure 6-1: JPEG encoding steps.......... 76

Figure 6-2: DCT-based decoding steps. 77

Figure 6-3: (a) Histogram of DC coefficients of a sample image and its Fourier transform (image compressed by a factor of 70); (b) Histogram of DC coefficients and its Fourier transform of an image which has been created by JPEG compressing the image (a) by the factor of 90.......... 79

Figure 6-4: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 ($QF_1 = 55$).......... 81

Figure 6-5: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 ($QF_1 = 70$).......... 82

Figure 6-6: Detection rates of the technique [13] for D-JPEG images (solid line) and cropped D-JPEG images (dashed line) for different QF_2 ($QF_1 = 85$).......... 82

Figure 6-7: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 55$. 84

Figure 6-8: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 70$. 84

Figure 6-9: Detection rates of [46] in detecting cropped D-JPEG images (solid lines) and attacked cropped D-JPEG images (dashed line) for different QF_2 and $QF_1 = 85$. 85

Figure 6-10: Shown on the left is the detection result of the JPEG image of Lena and on the right the detection result of the RD-JPEG version of the same image. 86

Figure 6-11: Detection results for RD-JPEG images by different scaling factors when the quality factor of the trained images and the test images are the same. 89

Figure 6-12: Detection results for RD-JPEG images by different scaling factors when the quality factor of the trained images and the test images are different. 89

List of Tables

Table 4-1: Detection rates, correctness rates and incorrectness rates while evaluating forged images where copied parts are square.......... 43

Table 4-2: Detection rates, correctness rates and incorrectness rates while evaluating forged dataset where copied parts are non-square. 43

Table 4-3: Detection rates and correctness rates after Gaussian noise addition.......... 44

Table 4-4: Detection rates and correctness rates after JPEG compression. 45

Table 4-5: Detection rates and correctness rates after rotation. 45

Table 4-6: Detection rates and correctness rates after changing LSB.......... 45

Table 4-7: Detection rates and correctness rates in evaluation for cropping and rescaling. 46

Table 4-8: Detection rates and correctness rates after swapping DCT coefficients.......... 46

Table 4-9: Detection rates and correctness rates after the combination attack of cropping, double JPEG compression and rescaling.......... 47

Table 4-10: Visual quality evaluation of attacked images by adding noise. 47

Table 4-11: Visual quality evaluation of attacked images by JPEG compressing. 47

Table 4-12: Visual quality evaluation of several attacks.......... 47

Table 4-13: Detection rates for detecting rotated images ($B = 16$). 52

Table 4-14: Detection rates for detecting forged images with added Gaussian noise ($B = 16$). 52

Table 4-15: Detection rates for detecting copy-rotate-moved images ($B = 16$).......... 52

Table 4-16: Detection rates for detecting rotated images ($B = 8$). 53

Table 4-17: Detection rates for detecting images with added Gaussian noise ($B = 8$).......... 53

Table 4-18 Detection rates for detecting copy-rotated-moved images ($B = 8$). 53

Table 5-1: Detection rates for up-sampled images after attack by using Gaussian noise, median filter, and order-statistic filter.......... 62

Table 5-2: Average PSNR between re-sampled images and attacked re-sampled images....... 63

Table 5-3: Detection rates when applying different attacks to up-sampled images by a scaling factor of 120%. 69

Table 5-4: Detection rates for up-sampled images by a factor of 20% for and added Gaussian noise with SNR of 20 dB, 25 dB, 30 dB, and 35 dB. 71

Table 5-5: Difference between re-sampled images and attacked re-sampled images.......... 71

Table 6-1: Detection rates of [13] in detecting double JPEG compressed images (QF_1 in rows and QF_2 in columns) [%].......... 80

Table 6-2: Detection results using a single SVM classifier (training JPEG images compressed by QF = 50 and RD-JPEG images re-compressed by $QF_1 = 50$, $QF_2 = 70$) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns). 87

Table 6-3: Detection results using a single SVM classifier (training JPEG images compressed by QF = 70 and RD-JPEG images re-compressed by $QF_1 = 70$, $QF_2 = 80$) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns). 88

Table 6-4: Detection results using a single SVM classifier (training JPEG images compressed by QF=70 and RD-JPEG images re-compressed by QF_1=70, QF_2=80) for RD-JPEG images by the scaling factor of 1.1 and by different quality factors (QF_1 in rows and QF_2 in columns). 88

Table 6-5: Detection results using dedicated SVM classifiers for RD-JPEG images (depending on the quality factor of the second compression) by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns). 88

Table 8-1: The standard quantization table 93

List of Abbreviations

ASA	Advertising Standard Authority
BACM	Blocking Artifact Characteristic Matrix
CCD	Charge-Coupled Device
CFA	Color Filter Array
CMOS	Complimentary Metal-Oxide Semiconductor
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DIP	Digital Image Processing
D-JPEG	Double JPEG
DR	Detection Rate
DWT	Discrete Wavelet Transform
EM	Expectation/Maximization
EZW	Embedded Zero-tree Wavelet
FMT	Fourier-Mellin Transform
FPR	False Positive Rate
HMF	Hybrid Median Filter
ICR	Incorrectness Rate
IDCT	Inverse DCT
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
NVF	Noise Visibility Function
PC	Phase Correlation
PCA	Principal Component Analysis
PRNU	Photo-Response Non-Uniformity
PSNR	Peak Signal to Noise Ratio
QF	Quality Factor
RD-JPEG	Resized Double JPEG
RT	Radon Transformation
RTPC	Radon Transform Phase Correlation
SIFT	Scale Invariant Features Transform
SNR	Signal to Noise Ratio
SSIM	Structural Similarity Index Metric
SPIHT	Set Partitioning in Hierarchical Trees

SURF	Speed Up Robust Features
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TPR	True Positive Rate
UCID	Uncompressed Color Image Database
WLS	Weight Least Square
WPSNR	Weight Peak Signal to Noise Ratio

Bibliography

[1] H. T. Sencar and N. Memon, “Overview of State-of-the-Art in Digital Image Forensics,” *In Proc. WSPC*, 2007.

[2] S. Katzenbeisser and F. Petitcolas, “Information Hiding Techniques for Steganography and Digital Watermarking,” *Artech House*, 2001.

[3] H. Liu, “Digital Watermarking for Content Authentication,” *PhD Thesis, TU Darmstadt*, 2008.

[4] “Photo Tampering throughout History.” [Online]. Available: http://www.fourandsix.com/photo-tampering-history/.

[5] “Nature International Weekly Journal of Science.” [Online]. Available: http://www.nature.com/news/2009/091021/full/4611035a.html.

[6] “Huffingtonpost.” [Online]. Available: http://www.huffingtonpost.com/2012/10/23/natalie-portman-dior-ad-banned-mascara_n_2004837.html#slide=2321455.

[7] A. Piva, “An Overview on Image Forensics,” *ISRN Signal Processing*, vol. 2013, pp. 1–22, 2013.

[8] J. Adams, K. Parulski, and K. Spaulding, “Color Processing in Digital Cameras,” *IEEE Micro*, vol. 18, no. 6, pp. 20–30, 1998.

[9] T. Van Lanh, K.-S. Chong, S. Emmanuel, and M. S. Kankanhalli, “A Survey on Digital Camera Image Forensic Methods,” *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 16–19, Jul. 2007.

[10] M. Kharrazi, H. Sencar, and N. Memon, “Blind Source Camera Identification,” *ICIP*, pp. 2–5, 2004.

[11] M. Tsai, G. Wu, I. Management, and N. Chiao, “Using image features to identify camera sources,” *ICASSP*, pp. 297–300, 2006.

[12] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, “Source Camera Identification Using Footprints from Lens Aberration,” *SPIE-IS&T*, vol. 6069, no. 852, pp. 1–8, 2006.

[13] A. Popescu, “Statistical Tools for Digital Image Forensics,” *PhD Thesis*, 2004.

[14] S. Bayram, H. T. Sencar, and N. Memon, “Source Camera Identification Based on CFA Interpolation,” *ICIP*, 2005.

[15] A. Swaminathan, S. Member, M. Wu, S. Member, and K. J. R. Liu, “Nonintrusive Component Forensics of Visual Sensors Using Output Images,” *ICIP*, vol. 2, no. 1, pp. 91–106, 2007.

[16] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, “Methods for identification of images acquired with digital cameras,” *Proc. SPIE 4232, Enabling Technologies for Law Enforcement and Security*, pp. 505–512, Feb. 2001.

[17] J. Janesick, *Scientific Charge-Coupled Devices*. SPIE Press, 2001.

[18] J. Fridrich, D. Soukal, and J. Lukas, “Detection of Copy-Move Forgery in Digital Images,” *Proc. of Digital Forensic Research Workshop*, Dec. 2003.

[19] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, “Determining Image Origin and Integrity Using Sensor Noise,” *IEEE Trans. on Information Forensics and Security*, 2008.

[20] J. Lukás, J. Fridrich, and M. Goljan, “Digital Camera Identification From Sensor Pattern Noise,” *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[21] C.-T. Li and Y. Li, “Digital camera identification using Colour-Decoupled photo response non-uniformity noise pattern,” *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 3052–3055, May 2010.

[22] B. Liu, Y. Hu, and H. Lee, “Department of Electrical Engineering and Computer Science Korea Advanced Institute of Science and Technology , Republic of Korea School of Electronic and Information Engineering,” *ICIP*, no. 1, pp. 1749–1752, 2010.

[23] C. Li, “Source Camera Identification Using Enhanced Sensor Pattern Noise,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, Jun. 2010.

[24] M. Goljan and J. Fridrich, “Sensor-Fingerprint Based Identification of Images Corrected for Lens Distortion,” *Proc. of SPIE: Media Watermarking, Security, and Forensics*, vol. 8303, 2012.

[25] C. McKay, A. Swaminathan, and M. Wu, “Image acquisition forensics: Forensic analysis to identify imaging source,” *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1657–1660, Mar. 2008.

[26] “Gizmodo.” [Online]. Available: http://gizmodo.com/238760/computer+generated-image-looks-exactly-like-a-beautiful-woman.

[27] S. Lyu and H. Farid, “How realistic is photorealistic?,” *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 845–850, Feb. 2005.

[28] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, “Physics-motivated features for distinguishing photographic images and computer graphics,” *Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA ’05*, p. 239, 2005.

[29] Y. Wang and P. Moulin, “On Discrimination Between Photorealistic and Photographic Images,” *ICASSP*, 2006.

[30] S. Dehnie, T. Sencar, and N. Memon, “Digital Image Forensics for Identifying Computer Generated and Digital Camera Images,” *ICIP*, 2006.

[31] E. Dirik, S. Bayram, H. T. Sencar, and N. Memon, “New Features to Identify Computer Generated Images,” *ICIP*, 2007.

[32] H. Farid, “Image Forgery Detection,” *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[33] S. Khan and A. Kulkarni, “Reduced Time Complexity for Detection of Copy-Move Forgery Using Discrete Wavelet Transform,” *IJCA*, vol. 6, no. 7, pp. 31–36, 2010.

[34] S. Bayram and H. T. Sencar, “An efficient and robust method for detection copy-move forgery,” *Image (Rochester, N.Y.)*, 2008.

[35] A. Popescu and H. Farid, “Exposing Digital Forgeries by Detecting Duplicated Image Regions,” *TR2004-515, Dartmouth College, Computer Science*, 2006.

[36] H. Farid, “Detecting Digital Forgeries Using Bispectral Analysis,” *AI Lab, MIT Technical Report*, 1999.

[37] T.-T. Ng and S.-F. Chang, “A Model for Image Splicing,” *ICIP*, 2004.

[38] A. C. Popescu and H. Farid, “Exposing digital forgeries by detecting traces of resampling,” *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[39] M. Kirchner, “Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue,” *Proceedings of the 10th ACM workshop on Multimedia and security - MM&Sec ’08*, p. 11, 2008.

[40] A. C. Gallagher, “Detection of Linear and Cubic Interpolation in JPEG Compressed Images,” *The 2nd Canadian Conference on Computer and Robot Vision (CRV’05)*, no. 2, pp. 65–72, 2005.

[41] B. Mahdian and S. Saic, “Blind Authentication Using Periodic Properties of Interpolation,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, Sep. 2008.

[42] J. Lukáš and J. Fridrich, “Estimation of Primary Quantization Matrix in Double Compressed JPEG Images,” *In Proc. Digital Forensic Research Workshop*, 2003.

[43] A. C. Popescu and H. Farid, “Statistical Tools for Digital Forensics,” *in Proc. 6th Int. Workshop Information Hiding*, 2004.

[44] W. Luo, Z. Qu, J. Huang, and G. Qiu, “A Novel Method for Detecting Cropped and Recompressed Image Block,” *IEEE ICASSP 2007*, pp. 217–220, 2007.

[45] Y. Chen and C. Hsu, “Image Tampering Detection by Blocking Periodicity Analysis in JPEG Compressed Images,” *MMSP 2008*, no. c, pp. 803–808, 2008.

[46] T. Bianchi and A. Piva, “Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps,” *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2012.

[47] A. Popescu and H. Farid, “Exposing digital forgeries in color filter array interpolated images,” *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, Oct. 2005.

[48] A. Dirik and N. Memon, “Image Tamper Detectiong Based on Demosaicing Artifacts,” *ICIP*, pp. 1497–1500, 2009.

[49] P. Ferrara, T. Bianchi, A. De Rosa, A. Piva, and S. Member, “Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts,” *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[50] M. K. Johnson and H. Farid, “Exposing digital forgeries through chromatic aberration,” *Proceeding of the 8th workshop on Multimedia and security - MM&Sec ’06*, no. 2, p. 48, 2006.

[51] M. Chen, J. Fridrich, J. Lukáš, and M. Goljan, “Imaging Sensor Noise as Digital X-Ray for Revealing Forgeries,” *In Proc 9th Int. Workshop on Statistical Analysis in Computer Vision*, 2007.

[52] M. K. Johnson and H. Farid, “Exposing digital forgeries by detecting inconsistencies in lighting,” *Proceedings of the 7th workshop on Multimedia and security - MM&Sec ’05*, pp. 1–10, 2005.

[53] M. K. Johnson and H. Farid, “Exposing Digital Forgeries Through Specular Highlights on the Eye,” *IH*, 2007.

[54] M. K. Johnson and H. Farid, “Exposing Digital Forgeries in Complex Lighting Environments,” *IEEE Trans. on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, 2007.

[55] J. Zhu and P. Wang, “Detecting Photographic Composites Using Shadows,” *ICME*, pp. 1042–1045, 2009.

[56] Q. Liu, X. Cao, C. Deng, and X. Guo, “Identifying Image Composites Through Shadow Matte Consistency,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1111–1122, Sep. 2011.

[57] M. K. Johnson and H. Farid, “Detecting Photographic Composites of People,” *IWDW*, 2007.

[58] P. Kakar, N. Sudha, and W. Ser, “Exposing Digital Image Forgeries by Detecting Discrepancies in Motion Blur,” *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 443–452, 2011.

[59] B. Rainer and M. Kirchner, *Counter-Forensics : Attacking Image Forensics*. Springer, 2012.

[60] M. Kirchner and R. Boehme, “Hiding Traces of Resampling in Digital Images,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, Dec. 2008.

[61] H. C. Nguyen and S. Katzenbeisser, “Robust Resampling Detection in Digital Images,” *CMS 2012*, pp. 3–15, 2012.

[62] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, “Can we trust digital image forensics?,” *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA ’07*, p. 78, 2007.

[63] M. Kirchner and R. Böhme, “Synthesis of color filter array pattern in digital images,” *Proceedings of SPIE*, no. 2, p. 72540K–72540K–14, 2009.

[64] M. Stamm, S. Tjoa, S. Lin, K. J. R. Liu, and J. Liu, “Undetectable image tampering through JPEG compression,” *ICIP 2010*, 2010.

[65] M. Stamm and K. J. R. Liu, “Wavelet-based Image Compression Anti-Forensics,” *ICIP*, 2010.

[66] M. Kirchner and J. Fridrich, “On detection of median filtering in digital images,” *Computer*, pp. 754110–754110–12, 2010.

[67] G. Valenzise, M. Tagliasacchi, and S. Tubaro, “The Cost of JPEG Compression Anti-Forensics,” *ICASSP*, pp. 1884–1887, 2011.

[68] M. Wirth, M. Fraschini, M. Masek, and M. Bruynooghe, “Performance Evaluation in Image Processing,” *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–4, 2006.

[69] F. A. P. Petitcolas, “Stirmark,” 1999. [Online]. Available: http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/index.html.

[70] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Mailler, and T.Pun, “Checkmark,” 2001. [Online]. Available: http://watermarking.unige.ch/checkmark/.

[71] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, and I. Pitas, “Optimark,” 2001. [Online]. Available: http://poseidon.csd.auth.gr/optimark/.

[72] N. Nikolaidis, V. Solachidis, A. Tefas, V. Arguriou, and I. Pitas, “Benchmarking of still image watermarking methods: principles and state of the art,” *Processing*, 2001.

[73] S. Bayram and H. T. Sencar, “A survey of copy-move forgery detection techniques,” *Proc. IEEE Western New York Image Processing Workshop, Rochester, NY, USA*, 2008.

[74] H. C. Nguyen and S. Katzenbeisser, “Performance and Robustness Analysis for Some Resampling Detection Techniques in Digital Images,” in *IWDW*, 2011.

[75] V. Christlein, C. Riess, and E. Angelopoulou, “A Study on Features for the Detection of Copy-Move Forgeries,” *GI Sicherheit*, 2010.

[76] S. Battiato, V. A. Doria, and G. Messina, “Digital Forgery Estimation into DCT Domain - A Critical Analysis,” *ACM MiFor*, pp. 0–5, 2009.

[77] H. C. Nguyen and S. Katzenbeisser, “Security of Copy-Move Forgery Detection Techniques,” *ICASSP*, pp. 1864–1867, 2011.

[78] Z. Wang and A. C. Bovik, “Modern Image Quality Assessment,” *Morgan & Claypool*, 2006.

[79] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity.,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 13, no. 4, pp. 600–12, Apr. 2004.

[80] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun, “A Stochastic Approach to Content Adaptive Digital Image Watermarking,” *International Workshop on Information Hiding*, 1999.

[81] M. Kutter and F. Petitcolas, “Fair benchmark for image watermarking systems,” *Proceedings of SPIE*, vol. 3657, no. January, pp. 226–239, 1999.

[82] G. Schaefer and M. Stich, “UCID: an uncompressed color image database,” *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose, USA*, pp. 472–480, 2004.

[83] M. K. Bashar, K. Noda, N. Ohnishi, H. Kudo, T. Matsumoto, and Y. Takeuchi, “Wavelet-Based Multiresolution Features for Detecting Duplications in Images,” *Conference on Machine Vision Applications*, pp. 2–5, 2007.

[84] X. Kang and S. Wei, “Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics,” *2008 International Conference on Computer Science and Software Engineering*, pp. 926–930, 2008.

[85] W. Luo, J. Huang, and G. Qiu, “Robust Detection of Region-Duplication Forgery in Digital Image,” *18th International Conference on Pattern Recognition (ICPR’06)*, pp. 746–749, 2006.

[86] B. Mahdian and S. Saic, “Detection of copy-move forgery using a method based on blur moment invariants.,” *Elsevier Forensic Science International*, vol. 171, no. 2–3, pp. 180–189, Sep. 2007.

[87] S. Ryu, M. Lee, and H. Lee, “Detection of Copy-Rotate-Move Forgery Using Zernike Moments,” *IH 2010, LNCS 6387*, vol. 1, pp. 51–65, 2010.

[88] S. Bravo-solorio and A. K. Nandi, “Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling,” *EUSIPCO*, no. Eusipco, pp. 824–828, 2009.

[89] V. Christlein, C. Riess, and E. Angelopoulou, “On rotation invariance in copy-move forgery detection,” *WIFS*, 2010.

[90] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[91] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF : Speeded Up Robust Features,” *ECCV 2006*, 2006.

[92] H. Huang, W. Guo, and Y. Zhang, “Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm,” *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp. 272–276, Dec. 2008.

[93] X. Pan and S. Lyu, “Detecting Image Region Duplication Using SIFT Features,” *ICASSP 2010*, pp. 1706–1709, 2010.

[94] I. Amerini, L. Ballan, S. Member, R. Caldelli, A. Del Bimbo, and G. Serra, “A SIFT-based forensic method for copy-move attack detection and transformation recovery,” *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 1, pp. 1–12, 2011.

[95] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, “Image Copy-Move Forgery Detection Based on SURF,” *2010 International Conference on Multimedia Information Networking and Security*, pp. 889–892, 2010.

[96] E. Koch and J. Zhao, “Towards Robust and Hidden Image Copyright Labeling,” *Proc. IEEE Workshop on nonlinear signnal and image processing*, pp. 20–23, 1995.

[97] B. Jähne, *Digital Image Processing*, 6th ed. Springer, 2005, p. 237.

[98] Y.-G. Wang, Y. Lei, and J. Huang, “An image copy detection scheme based on radon transform,” *2010 IEEE International Conference on Image Processing*, pp. 1009–1012, Sep. 2010.

[99] R. Galiegekere, D. Holdsworth, and A. Fenster, “Moment pattern in the Radon space,” *Society of Photo-optical instrumentation engineering*, vol. 39, no. 4, pp. 1088–1097, 2000.

[100] K. Jafari-Khouzani and H. Soltanian-Zadeh, “Rotation-invariant multiresolution texture analysis using radon and wavelet transforms.,” *IEEE transactions on image processing*, vol. 14, no. 6, pp. 783–95, Jun. 2005.

[101] G. Wolberg, “Digital Image Warping,” *IEEE Computer Society Press Los Alamitos, CA, USA*, 1994.

[102] M. Kirchner, “Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue,” *Proceedings of the 10th ACM Workshop on Multimedia and Security - MM&Sec’08*, 2008.

[103] M. Kirchner and T. Gloe, “On resampling detection in re-compressed images,” *WIFS*, pp. 21–25, 2009.

[104] S. Prasad and K. R. Ramakrishnan, “On resampling detection and its application to detect image tampering,” *ICME*, 2006.

[105] F. Uccheddu, A. De Rosa, A. Piva, and M. Barni, “Detection of resampled images: performance analysis and practical challenges,” *EURASIP*, pp. 1675–1679, 2010.

[106] B. Mahdian and S. Saic, “http://zoi.utia.cas.cz/files/rsmp_core.txt.”

[107] A. Dempster, N. Laird, and D. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[108] D. Garcia, “BiomeCardio.” [Online]. Available: http://www.biomecardio.com/matlab/hmf.html.

[109] B. Mahdian and S. Saic, “On Periodic Properties of Interpolation and Their Application To Image Authentication,” *Third International Symposium on Information Assurance and Security*, pp. 439–446, Aug. 2007.

[110] Z. Lin, J. He, X. Tang, and C.-K. Tang, “Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis,” *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.

[111] T. Bianchi, A. De Rosa, and A. Piva, “Improved DCT coefficient analysis for forgery localization in JPEG images,” *ICASSP 2011*, pp. 2444–2447, 2011.

[112] S. Ye, Q. Sun, and E.-C. Chang, “Detecting digital image forgeries by measuring inconsistencies of blocking artifact,” *ICME*, vol. 117543, no. 2, pp. 12–15, 2007.

[113] D. Fu, Y. Q. Shi, and W. Su, “A generalized Benford’s law for JPEG coefficients and its applications in image forensics,” *Proceedings of SPIE*, p. 65051L–65051L–11, 2007.

[114] B. Li, Y. Q. Shi, and J. Huang, “Detecting doubly compressed JPEG images by using Mode Based First Digit Features,” *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pp. 730–735, Oct. 2008.

[115] H. Farid, “Exposing Digital Forgeries From JPEG Ghosts,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.

[116] C. Chen, Y. Q. Shi, and W. Su, “A machine learning based scheme for double JPEG compression detection,” *2008 19th International Conference on Pattern Recognition*, pp. 1–4, Dec. 2008.

[117] P. Sutthiwan and Y. Q. Shi, “Anti-Forensics of Double JPEG Compression Detection,” *IWDW*, pp. 411–424, 2011.

[118] Q. Liu and A. H. Sung, “A new approach for JPEG resize and image splicing detection,” *Proceedings of the First ACM workshop on Multimedia in forensics - MiFor ’09*, p. 43, 2009.

[119] T. Bianchi and A. Piva, “Reverse engineering of double JPEG compression in the presence of image resizing,” *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, no. 1, pp. 127–132, Dec. 2012.

[120] G. K. Wallace, “The JPEG Still Picture Compression Standard,” *IEEE Transactions on Consumer Electronics*, pp. 1–17, 1991.

[121] F. Huang, J. Huang, S. Member, and Y. Q. Shi, “Detecting Double JPEG Compression With the Same Quantization Matrix,” *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 4, pp. 848–856, 2010.

[122] M. C. Stamm and K. J. R. Liu, “Anti-Forensics of Digital Image Compression,” *IEEE Trans. on Information Forensics and Security*, no. c, 2010.

[123] J. Kornblum, “Using JPEG quantization tables to identify imagery processed by software,” *Digital Investigation*, vol. 5, pp. S21–S25, Sep. 2008.

[124] S. Chandra and C. S. Ellis, “JPEG Compression Metric as a Quality Aware Image Transcoding,” *Proceedings of USITS’ 99*, 1999.

[125] P. Sallee, “Matlab JPEG Toolbox.” [Online]. Available: http://www.philsallee.com/jpegtbx/index.html.

Wissenschaftlicher Werdegang

Nguyen Hieu Cuong

1991 – 1995	Studium der Mathematik und Informatik (Bachelor) an der Universität Hanoi, Vietnam
1996 – 1999	Studium der Mathematik und Informatik (Master) an der Universität Hanoi, Vietnam
1995 – jetzt	Wissenschaftlicher Mitarbeiter und Dozent an der Vehrkehrshochschule Hanoi, Vietnam
Dezember 1997 – August 1998	Praktikum im Institut für Numerische Methoden und Informatik im Bauwesen an der TU Darmstadt
Oktober 2009 – September 2013	Promotion an der TU Darmstadt unter Leitung von Prof. Dr. Stefan Katzenbeisser