ABSTRACT

Title of Dissertation: MULTIMEDIA FORENSIC ANALYSIS VIA INTRINSIC AND EXTRINSIC FINGERPRINTS

Ashwin Swaminathan, Doctor of Philosophy, 2008

Dissertation directed by: Professor Min Wu
Department of Electrical and Computer Engineering

Digital imaging has experienced tremendous growth in recent decades, and digital images have been used in a growing number of applications. With such increasing popularity of imaging devices and the availability of low-cost image editing software, the integrity of image content can no longer be taken for granted. A number of forensic and provenance questions often arise, including how an image was generated; from where an image was from; what has been done on the image since its creation, by whom, when and how. This thesis presents two different sets of techniques to address the problem via *intrinsic* and *extrinsic* fingerprints.

The first part of this thesis introduces a new methodology based on *intrinsic fingerprints* for forensic analysis of digital images. The proposed method is motivated by the observation that many processing operations, both inside and

outside acquisition devices, leave distinct intrinsic traces on the final output data. We present methods to identify these *intrinsic fingerprints* via component forensic analysis, and demonstrate that these traces can serve as useful features for such forensic applications as to build a robust device identifier and to identify potential technology infringement or licensing.

Building upon component forensics, we develop a general authentication and provenance framework to reconstruct the processing history of digital images. We model post-device processing as a manipulation filter and estimate its coefficients using a linear time invariant approximation. Absence of in-device fingerprints, presence of new post-device fingerprints, or any inconsistencies in the estimated fingerprints across different regions of the test image all suggest that the image is not a direct device output and has possibly undergone some kind of processing, such as content tampering or steganographic embedding, after device capture.

While component forensics is widely applicable in a number of scenarios, it has performance limitations. To understand the fundamental limits of component forensics, we develop a new theoretical framework based on estimation and pattern classification theories, and define formal notions of forensic identifiability and classifiability of components. We show that the proposed framework provides a solid foundation to study information forensics and helps design optimal input patterns to improve parameter estimation accuracy via semi non-intrusive forensics.

The final part of the thesis investigates a complementing extrinsic approach via image hashing that can be used for content-based image authentication and other media security applications. We show that the proposed hashing algorithm is robust to common signal processing operations and present a systematic evaluation of the security of image hash against estimation and forgery attacks.

MULTIMEDIA FORENSIC ANALYSIS VIA INTRINSIC AND EXTRINSIC FINGERPRINTS

by

Ashwin Swaminathan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:

Professor Min Wu, Chair / Advisor
Professor K. J. Ray Liu
Professor Alexander Barg
Professor Adrian Papamarcou
Professor Douglas W. Oard

DEDICATION

To my parents and my sister.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Prof. Min Wu, for her guidance and support during my graduate study at the University of Maryland. She has always encouraged me to be novel and engage in creative thinking and reasoning. Her patience, never-say-die attitude, and perseverance have inspired me to work harder and strive for the best. She has constantly taken an active part in steering me to explore the depths and the breadths of my research and has guided me through its several challenges. Her mentoring has been paramount in improving the outcome of my research and her guidance has also impacted and shaped my skills as a researcher. As my Ph.D. draws to a close and my career as a researcher begins, I am confident that these very traits that I have imbibed will aid me in my future endeavors.

I also appreciate Prof. K. J. Ray Liu for his invaluable advice and guidance during my Ph.D. His vision and insights have been a constant guiding factor during several stages of my Ph.D. He helped me formulate the right problems and address the right questions. He has always encouraged me to think outside the box and approach research from multiple perspectives. Additionally, I am also indebted to Dr. Ton Kalker and Dr. Darko Kirovski for their encouragement, mentoring, and research support during my internship at Hewlett-Packard Labs and Microsoft Research, respectively. Both experiences were mentally stimulating and rich in their learning experience.

Further, I would like to thank Prof. Alexander Barg, Prof. Adrian Papamarcou, and Prof. Douglas Oard for their valuable comments on my thesis and serving on my dissertation committee. I am also grateful to Prof. Rama Chellappa for his help and support during my graduate studies and his course on Pattern Classification which was very useful for my Ph.D. work.

I would also like to thank my colleagues, collaborators, and office-mates at the University of Maryland: Dr. Yinian Mao, Dr. Shan He, Dr. Hongmei Gou, Dr. Guan-Ming Su, Avinash Varna, Wan-Yi Lin, Wenjun Lu, Wei-Hong Chuang, Prof. Hong Zhao, Dr. Meng Chen, Dr. Ahmed Sadek, Steve Tjoa, Beibei Wang, and Matthew Stamm. I have enjoyed the close working environment. I would also like to thank Prof. Nasir Memon and Prof. Kivanç Mihçak for their valuable comments and suggestions on Chapter 7 of the thesis.

Many thanks to my friends: Mahesh Ramachandran, Aswin Sankaranarayan, and Ashok Veeraraghavan who in different stages of my graduate study have helped me gain significant insights to my Ph.D. work. I am also very grateful to my house-mates: Krishna Raj, Ranjit Kumaresan, Sebastian Thomas, Charles Tobin, Thangamani Veeramani, Kumar Ravichandran, and Sidharth Kumar; and my friends: Alankar Bandyopadhyay, Jagan Sankaranarayanan, Rajesh Sathiyanarayanan, Eswaran Baskaran, Karthik Ravirajan, Arvind Sundaresan, Anuj Rawat, Narayanan Ramanathan, Gaurav Agarwal, Deepak Sridharan, Sandeep Manocha, Adarsh Sridhar, Balaji Vasan, Kiran Kumar, Bargava Raman, Bhargav Kanagal, Prashanth Ananthapadbhanaban, and Chandrasekhar Nagarajan. While the space here does not permit to list all the friends I made at UMD, I appreciate and cherish their friendship and help during my graduate study.

Finally, I wish to express my heartfelt gratitude to my parents and sister. I

thank my mother, Bhama Swaminathan, for my upbringing. Through her endless love and understanding, she instilled in me a sense of diligence and discipline, which have been vital during my Ph.D work. She emphasized the virtues of honesty, sincerity, and integrity. My father, P.V. Swaminathan, is the person who inspired my interests in engineering since childhood, and constantly encouraged me to pursue my dreams and focus on realizing my goals. My sister, Swetha Swaminathan, has always been my best friend and philosopher. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivations

Visual sensor technologies have experienced tremendous growth in recent decades. The resolution and quality of electronic imaging has been steadily improving, and digital cameras are becoming ubiquitous. Shipment of digital cameras alone has grown from $46.4 million in 2003 to $62 million in 2004, and this forms an approximately $15 billion market worldwide [5]. Digital images taken by various imaging devices have been used in a growing number of applications, from military and reconnaissance to medical diagnosis and consumer photography. Consequently, a series of new forensic issues arise amidst such rapid advancement and widespread adoption of imaging technologies. For example, one can readily ask what kinds of hardware and software components as well as their parameters have been employed inside the devices? Given a digital image, which imaging sensor or which brand of sensors was used to acquire the image? What kinds of legitimate processing and undesired alteration have been applied to an image since it leaves the device? How would you authenticate such device captured images?

Some of these forensic questions are related to identifying the source of the digital image, and determining possible tampering or presence of hidden data. Evidence obtained from such forensic analysis would provide useful forensic information to law enforcement and intelligence agencies as to if the given image was actually captured with a camera (or generated by other means) and to establish the authenticity of the digital image. In this thesis, we present two different approaches to address this problem based on *intrinsic* and *extrinsic* fingerprints.

Intrinsic fingerprints are internal traces left behind on the final digital image by the image capturing device. Each digital device can be broken into a number of its internal components, each performing a particular role. When the device is used to take a picture, the information of the real-world scene passes through the digital device and through each of its internal components before the final image is formed. Each of these components in the digital device modifies the input scene via a particular algorithm and leaves some *intrinsic* fingerprint traces on the final output. In this thesis, we develop a new forensic methodology called *component forensics*, which aims at identifying the intrinsic fingerprints left behind by each component inside a visual device by inferring what algorithms/processing are employed and estimating their parameter settings. Building upon component forensics, we extend these ideas to address a number of larger forensic issues in discovering technology infringement, protecting intellectual property rights, and identifying acquisition devices.

For centuries, intellectual property protection has played a crucial role in fostering innovation, as it has been known for "adding the fuel of interest to the fire of genius" since the time of Abraham Lincoln. Fierce competition in the electronic imaging industry has led to an increasing number of infringement cases filed

in U.S. courts. The remunerations awarded to successful prosecution have also grown tremendously, sometimes in billions of dollars. For example, the Ampex Corporation has more than 600 patents related to digital cameras; and based on one of the patents it has received more than $275-million compensation from lawsuits and settlements involving patent infringement cases with many digital camera vendors [4].

According to the U.S. patent law [1], *infringement of a patent* consists of the unauthorized making, using, offering for sale or selling any patented invention during the term of its validity. Patent infringement is considered one of the most difficult to detect, and even harder to prove in the court of law. The burden of proof often lies on patent holders, who are expected to provide solid evidence to substantiate their accusations. A common way to perform infringement analysis is to examine the design and implementation of a product and to look for similarities with what have been claimed in existing patents, through some type of reverse engineering. However, this approach could be very cumbersome and ineffective. For example, it may involve going over VHDL design codes of an IC chip in charge of core information processing tasks, which is a daunting task even to the most experienced expert in the field. Such analysis is often limited to the *implementation of an idea* rather than the *idea* itself, and thus could potentially lead to misleading conclusions [93, 144]. Component forensics is an important methodology to detect patent infringement and protect intellectual property rights, by obtaining evidence about the algorithms employed in various components of the digital device.

Component forensics also serves as a foundation to establish the trustworthiness of imaging devices [131]. With the fast development of tools to manipulate multimedia data, the integrity of both content and acquisition device has become

particularly important when images are used as critical evidence in journalism, reconnaissance, and law enforcement applications. For example, information about hardware/software modules and their parameters in a camera can help in building *camera identification systems*. Such systems would provide useful *acquisition forensic* information to law enforcement and intelligence agencies about which camera or which brand of camera is used to acquire an image. Additionally, component forensics helps establish a solid model on the characteristics of images obtained directly from a camera. This in turn will facilitate *tampering forensics* to determine if there has been any additional editing and processing applied to an image after it has been captured by the camera.

We can classify component forensics into three main categories based on the nature of the available evidence:

1. **Intrusive Forensics:** A forensic analyst has access to the device in question and can disassemble it to carefully examine every part, including analyzing any available intermediate signals and states to identify the algorithms employed in its processing blocks.

2. **Semi Non-Intrusive Forensics:** An analyst has access to the device as a black box. He/she can design appropriate inputs to be fed into the device so as to collect forensic evidence about the processing techniques and parameters of the individual components inside.

3. **Non-Intrusive Forensics:** An analyst does not have access to the device in question. He/she is provided with some sample data produced by the device, and studies them to gather forensic evidence.

The proposed research focuses on completely non-intrusive and semi non-intrusive

component forensics of visual sensors, while the suggested technologies can be extended to other types of acquisition models. As a new addition to the emerging field of *digital forensic engineering*, we propose a novel framework for analyzing technologies employed inside digital cameras solely on output images/videos, and develop a set of *forensic signal processing* algorithms to identify the parameters of such important camera components as color filter array, color interpolation, and white balancing. In the first part of this thesis, we show that successful development of the proposed intrinsic fingerprint methodologies offer a powerful framework and solutions to a large number of critical forensic issues.

The final part of this thesis addresses the problem of multimedia forensics via *extrinsic fingerprinting*. Extrinsic fingerprints are external signals that are added to the image by the device after the image has been captured. These external signals can then be used to establish the authenticity of digital data and determine possible tampering. Compared with non-intrusive forensic analysis via intrinsic fingerprints, the use of extrinsic fingerprints necessitates the presence of the device at hand as the fingerprint needs to be added at the time of image acquisition. While this requirement imposes some additional constraints on their applicability, extrinsic fingerprinting techniques help build a content-based image authentication scheme that is collision-resistant, robust to common signal processing operations, and secure against estimation and forgery attacks, as will be shown in the thesis.

1.2 Thesis Organization

This dissertation is organized as follows. In Chapter 2, we introduce a system model for digital imaging devices and identify the main components that go into the making of the digital device and formulate the problem.

Chapter 3 considers the problem of non-intrusive component forensics and proposes a set of forensic signal processing techniques to identify the algorithms and parameters employed in individual processing modules in digital devices. We show through detailed simulations that the proposed algorithms are robust to various kinds of postprocessing that may occur in the camera and demonstrate that the estimated intrinsic fingerprint traces can be employed to provide forensic evidence for patent infringement cases, intellectual property rights management, and technology evolution studies for digital media.

In Chapter 4, we propose a set of forensic signal processing techniques to verify whether a given digital image is an direct device output or not. We introduce a new formulation to study the problem of image authenticity based on the observation that each in-device and post-device processing operation leave some distinct intrinsic fingerprint traces on the final image. We model post-device processing as a linear shift-invariant system and estimate its coefficients using blind deconvolution. The absence of in-device fingerprints from a test image indicates that the test image is not a direct output of a digital device and is possibly generated by other image production processes. Any change or inconsistencies among the estimated in-device fingerprints, or the presence of new types of fingerprints suggest that the image has undergone some kind of processing after the initial capture, such as tampering or steganographic embedding.

Complementing the methods in Chapter 3 and 4 that identify the algorithms and parameters of various parts of the information processing chain, Chapter 5 presents the theoretical aspect of multimedia forensics to help understand its limitations. Using ideas from estimation and pattern classification theories, we define formal notions of identifiability of components in the information processing

chain. We show that the parameters of certain device components can be accurately identified only in controlled settings through semi non-intrusive forensics, while the parameters of some others can be computed directly from the available sample data via complete non-intrusive analysis.

We extend the theoretical framework to quantify and improve the accuracies and confidence in component parameter identification for several forensic applications. In Chapter 6, we specifically consider applications of the theoretical analysis to semi non-intrusive forensics. We assume the availability of the digital device; and introduce a forensic methodology to estimate the component parameters more accurately by devising good testing conditions and designing optimal input patterns. We experimentally verify that by careful choice of input and test conditions, semi non-intrusive forensics can provide much lower errors and higher accuracies in parameter estimation compared to completely non-intrusive forensics by better capturing the intrinsic fingerprint traces.

Chapter 7 explores using extrinsic fingerprints in image authentication and other media security applications. In this chapter, we develop a new algorithm for generating an image hash based on Fourier transform features and controlled randomization. We formulate the robustness of image hashing as a hypothesis testing problem and evaluate the performance under various image processing operations. We then introduce a general framework to study and evaluate the security of image hashing systems by quantifying its uncertainty in terms of differential entropy. We show that the proposed hash function can provide excellent tradeoffs between security and robustness. The dissertation is concluded in Chapter 8, with discussions on future perspectives.

Chapter 2

System Model and Problem Formulation

In this chapter, we introduce the system model for digital imaging devices and formulate the problem of multimedia forensics. For our work, we use visual sensors and images captured by devices employing these sensors for illustration, while the suggested techniques can be appropriately modified and extended to other types of acquisition models, and sensing technologies.[1]

2.1 Image Acquisition Model in Digital Cameras

Figure 2.1 shows the image capture model in digital cameras. As illustrated in the figure, light from a scene passes through a lens and optical filters, and is finally recorded by an array of sensors. Few consumer-level color cameras directly acquire full-resolution information for all three primary colors (usually red, green,

[1] In our ongoing work, we have extended the proposed forensic techniques for images produced by other acquisition sources such as scanners [54, 55] and cell phone cameras [94].

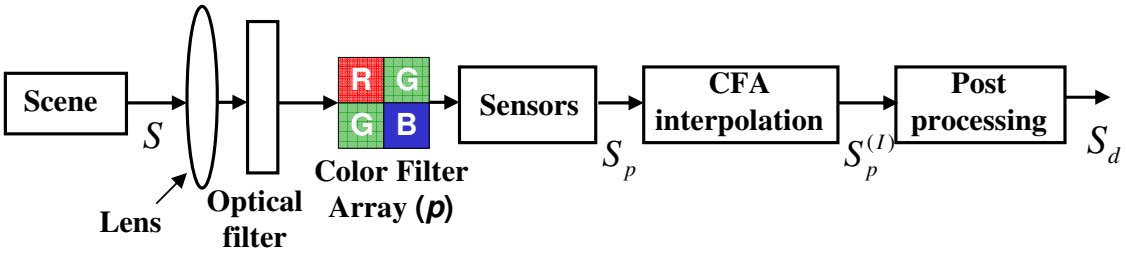


Figure 2.1: Image acquisition model in digital cameras.

and blue).[2] This is not only because of the high cost in producing a full-resolution sensor for each of the three colors, but also due to the substantial difficulty involved in perfectly matching the corresponding pixels and aligning the three color planes together. For these reasons, most digital cameras use a color filter array (CFA) to sample real-world scenes.

A color filter array consists of an array of color sensors, each of which captures the corresponding color of the real-world scene at an appropriate pixel location. Some examples of CFA patterns are shown in Figure 2.2. The Bayer pattern, shown in left corner of Figure 2.2, is one of the most popular CFA patterns. It uses a square lattice for the red and blue components of light and a diagonal lattice for the green color. The sensors are aligned on a square grid with the green color repeated twice compared to the corresponding red and blue sensors. The higher rate of sampling for the green color component enables to better capture the luminance component of light and thus provides better picture quality [6]. After CFA sampling, the remaining pixels are interpolated using the sampled data. Color interpolation (also known as demosaicking) is an important step to produce an output image with full resolution for all three color components [7, 112].

[2]New digital cameras employing Foveon X3 sensor, such as Sigma SD9 and Polaroid x530, capture all the three colors at each pixel location [2].

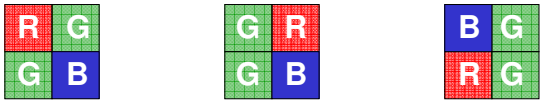


Figure 2.2: Sample color filter arrays.

To facilitate discussions, let S be the real-world scene to be captured by the camera and let p be the CFA pattern matrix. $S(x, y, c)$ can be represented as a 3-D array of pixel values of size $H \times W \times C$, where H and W represent the height and the width of the image, respectively, and $C = 3$ denotes the number of color components (red, green, and blue). The CFA sampling converts the real-world scene S into a three dimensional matrix S_p of the form

$$S_p(x, y, c) = \begin{cases} S(x, y, c) & \text{if } p(x, y) = c, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

After the data obtained from the CFA is recorded, the intermediate pixel values corresponding to the points where $S_p(x, y, c) = 0$ in (2.1) are interpolated using its neighboring pixel values to obtain $S_p^{(I)}$.

The performance of color interpolation directly affects the quality of the image captured by a camera [6,7,68]. There have been several commonly used algorithms for color interpolation. These algorithms can be broadly classified into two categories, namely, non-adaptive and adaptive algorithms. Non-adaptive algorithms apply the same type for interpolation for all pixels in a group. Some typical examples of non-adaptive algorithms include the nearest neighbor, bilinear, bicubic, and smooth hue interpolations [7]. Traditionally, the bilinear and bicubic interpolation algorithms are popular due to their simplicity and ease in hardware implementation. However, these methods are known to have significant blurring along edge regions due to averaging across edges. More computationally intensive adaptive

algorithms employing edge directed interpolation, such as the gradient based [92] and the adaptive color plane interpolation [56], have been proposed to reduce the blurring artifacts.

After interpolation, the three images corresponding to the red, green and the blue components go though a post-processing stage. In this stage, depending on the camera make and model, the images may undergo different processing operations [6, 7] which might include white balancing, color correction, gamma correction, lens vignetting correction, lens distortion removal, denoising, etc. Finally, the image may be JPEG compressed to reduce storage space to produce the output image S_d. For our work, we model all such post-interpolation processing as a combined post-processing block as shown in Figure 2.1.

2.2 Problem Formulation

In this thesis, we consider two approaches to multimedia forensics based on intrinsic and extrinsic fingerprints. These approaches are summarized in Figure 2.3 and Figure 2.4, respectively.

2.2.1 Forensic Analysis via Intrinsic Fingerprints

The system model for component forensics based on intrinsic fingerprint analysis is shown in Figure 2.3. As discussed in Chapter 1, the problem of component forensics deals with a methodology and systematic procedure to find the algorithms and parameters employed in various components in the device. Component forensics works by estimating the *intrinsic fingerprint* traces that are left behind in a digital image when it goes though various processing blocks in the information processing

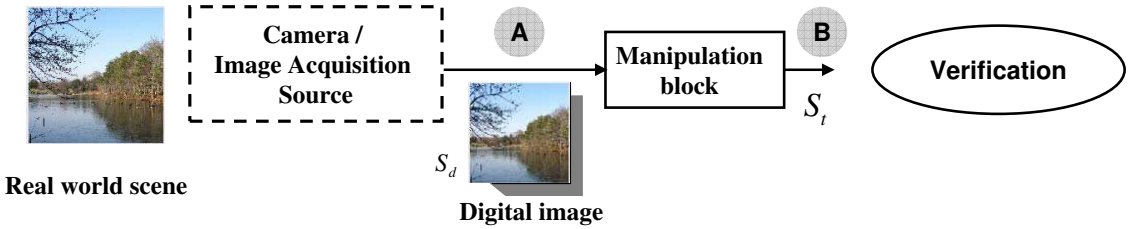


Figure 2.3: System model for intrinsic fingerprinting.

chain, and uses such traces for estimating component parameters. We classify the intrinsic fingerprint traces into two categories, namely, *in-camera* and *post-camera* fingerprints. Using a detailed imaging model, as described in Section 2.1, and its component analysis, we estimate the intrinsic fingerprints of the various *in-camera* processing operations. Specifically, we focus on such important camera components as color filter array and color interpolation and present methods to identify them based on the traces left behind on the final camera output (corresponding to the *point A* in Figure 2.3). The details of this work are presented in Chapter 3.

After the image has been produced by the camera, additional processing operations may be done using softwares such as Adobe Photoshop, Google Picasa, GIMP, *etc.* to further improve the picture quality and/or tamper with the image. In our system model, we represent such post-camera processing as an additional *manipulation block* as shown in Figure 2.3. Given the test image S_t, we assume that it is a manipulated camera output corresponding to the *point B* in Figure 2.3, and is obtained by processing the actual camera output S_d (*point A* in the figure) using the manipulation block. We introduce a two-step approach to detect post-camera manipulations. In the first step, we characterize the properties of a direct camera output using a camera model, and estimate its component parameters and the intrinsic fingerprints. We then represent the post-camera processing applied

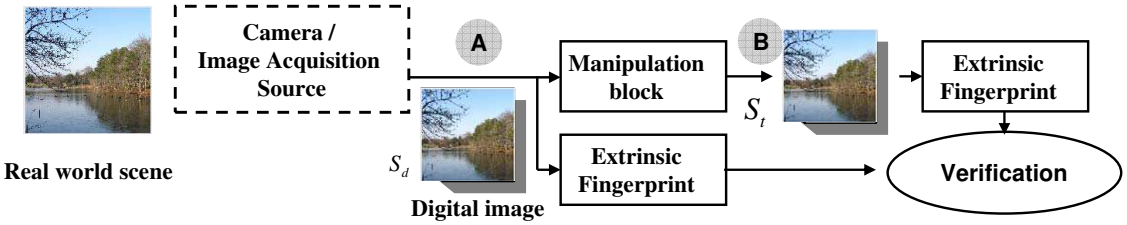


Figure 2.4: System model for extrinsic fingerprinting.

on S_d as a combination of linear and non-linear operations in the second step, and approximate them with a linear shift-invariant filter. The coefficients of this *manipulation filter*, estimated using blind deconvolution, serve as our post-camera fingerprints. In Chapter 4, we describe the estimation algorithm in detail.

2.2.2 Forensic Analysis via Extrinsic Fingerprints

As discussed in Chapter 1, extrinsic fingerprints are external signals added to the image by the camera after capture. They can be employed to establish the authenticity of images and determine possible tampering of hidden data. Figure 2.4 shows the system model for extrinsic fingerprinting. After the image has been captured by the camera, the camera inserts an extrinsic fingerprint, either in the form of a watermark embedded with the image or in the form of a hash appended along with the image. The image is then transmitted over the manipulation channel along with the extrinsic fingerprint. At the receiver end, the authenticator computes the extrinsic fingerprint of the manipulated image and compared them with the ones transmitted along with the data for verifying its authenticity. A high similarity among the estimated fingerprints from the manipulated image and the transmitted fingerprints suggests that the image has not undergone any manipulation after capture. On the other hand, a low similarity implies that image

has been manipulated via tampering or steganographic embedding operations. In this way, extrinsic fingerprints can help establish the authenticity of multimedia data. Chapter 7 will present details about the framework and design of extrinsic fingerprints.

Chapter 3

Non-Intrusive Component Forensics

In this chapter, we consider the problem of non-intrusive forensic analysis of digital cameras. We use sample images obtained from a digital camera under diverse and uncontrolled scene settings to determine the algorithms (and their parameters) employed in internal processing blocks. In particular, given an camera output image S_d (refer Figure 2.1), we focus on finding the color filter array pattern and the color interpolation algorithms, and show that the forensic analysis results of these components can be used as a first step in *reverse engineering* the making of a digital camera. The features and acquisition models that we develop in this chapter can be used to construct an efficient *camera identifier* that determines the brand/type of camera used to take the image. Further, our forensic algorithms can quantitatively help ascertain the similarities and differences among the corresponding camera components of different cameras. For devices from different vendors, the digital forensic knowledge obtained from such analysis can provide clues and evidence on technology infringement or licensing, which we shall refer to

as *infringement/licensing forensics* and will assist the enforcement of intellectual rights protection and foster technology innovation. For devices of the same brand but of different models released at different years and/or at various price tiers, our analysis forms a basis of *evolutionary forensics*, as it can provide clues on technology evolution. In the subsequent sections, we describe our proposed methodology and algorithms, and demonstrate their effectiveness with detailed simulation results and case studies. Later in Chapter 4, we show that the component forensic techniques can be employed to build a ground truth camera model to facilitate *tampering forensics*.

This chapter is organized as follows. We begin by reviewing prior work in non-intrusive forensic analysis in Section 3.1. In Section 3.2, we present methods to identify the CFA pattern and the color interpolation algorithm. We then illustrate proofs of concept with synthetic data in Section 3.3.1 and present results with a real data set of 19 cameras in Section 3.3.2. The estimated model parameters are used to construct a camera identifier and to study the similarities and differences among the cameras in Section 3.4. Section 3.5 generalizes the proposed methods to extend to other devices. The chapter is summarized in Section 3.6.

3.1 Related Work on Non-Intrusive Forensics

In literature, methods have been proposed to help identify the brand and model of the device just based on output data [10,14,15,21,26,69,70,70,75,83,112,136,136]. Choi *et al.* propose to employ the radial component of the lens distortion for camera identification [26] based on their hypothesis that the radial component varies among different camera models. The authors show through their simulation results that they can achieve a classification accuracy close to 91% over three different

camera models using this approach. In [70], Kharrazi *et al.* proposed a set of 34−features for camera identification aiming to model the image-capture process in digital cameras. The set of features include: average pixel value, RGB pairs correlation, neighbor distribution center of mass, RGB energy ratio, wavelet domain statistics [36], and image quality metrics [10]. The authors employ SVM for classification and report accuracies close to 88% when tested with pictures captured under controlled input conditions from five camera models of three different brands. The same set of features were also tested for camera identification in [136] where they report accuracies close to 95% over four different camera models from two different models again under controlled input conditions, and for cell phone camera identification in [21] with an accuracy close to 62.3% over 9 cell phone camera brands. These work do not target at explicitly estimating the various components of the information processing chain and only try to extract representative features for camera identification. Further, it is not clear as to which of these features enables identification, which might become very important in forensic investigations.

Chen and Hsu proposed a camera identification method based on camera gain histograms and features obtained from modelling camera noise to obtain an accuracy close to 85% over two camera models [25]. In [112], the authors employ Expectation/Maximization (EM) algorithms to estimate the color interpolation coefficients for forensic analysis. The authors first assume that the image pixels belong to one of the two hypothesis: (a) the pixel is linearly correlated to its neighbors and is obtained by a linear interpolation algorithm, and (b) the pixel is not correlated to its neighbors. Based on this assumption, the authors propose a two-step EM algorithm to estimate the CFA coefficients [112]. In the expectation step, the probability of each sample belonging to the two models is estimated, and

the specific form of the correlations is found in the Maximization step. The EM algorithm generates two outputs: a two-dimensional probability map indicating the likelihood of the pixel belonging to the two models and the weighting coefficients. Using these two outputs from the EM algorithm, Bayram *et al.* developed a camera identification method employing the weighting coefficients and the peak location and magnitudes of the frequency spectrum of the probability map as features [15]. Images captured from two cameras under controlled input conditions along with randomly acquired images from the Internet for the third camera were used for in the experiments, and the authors report accuracies close to 84% on three brands [15] when 20% of the 140 images were used in training and the remaining 80% employed in testing. Further improvements to this algorithm were made in [14] by separately considering smooth and non-smooth regions in the image to obtain accuracies close to 96% for three camera brands. Quadriatic pixel correlation model was used in [83] where the color interpolation coefficients were approximated by a linear model to give a classification accuracy close to 80%. Compared with these work on camera identification [14, 15, 70, 83, 136], the component forensics methodology described in this dissertation provides better discriminating power by doing a joint estimation of the CFA pattern and the interpolation algorithm.

Geradts *et al.* examine the effects of CCD pixels and used them to match images to the source camera [50]. Building upon these techniques Lukas *et al.* introduced a method for camera identification by estimating the pixel non-uniformity noise, which is a dominant component of the photo-response non-uniformity noise, inherent to an image sensor to distinguish between two cameras of the same brand, model, and set [85]. In the training phase of the algorithm, a wavelet based de-

noising algorithm is employed to obtain an estimate of the pixel non-uniformity noise and the random component of this noise is eliminated by averaging the estimates from a number of images. In the testing phase, to determine whether a given image is captured by a digital camera or not, the noise pattern from the image is obtained and correlated with the average noise pattern (also called the 'reference pattern') of the given digital camera. A correlation value greater than the pre-chosen threshold suggests that the given image is from the digital camera. The authors show that such an approach can identify the digital camera source with 100% accuracy when tested with high quality images. While useful in some forensic tasks when a suspicious camera is available for testing, this approach does not provide information about the internal components and cannot be used for identifying common features tied to the same camera models and brands.

Compared to these alternative approaches, the component forensic techniques introduced in our work are less dependent on input scenes and are robust against various common in-camera processing, and provide a high classification accuracy over a much larger database, as will be seen in as will be seen in Section 3.3.2.

3.2 Parameter Estimation of Camera Components

In this section, we develop a robust and non-intrusive algorithm to jointly estimate the CFA pattern and the interpolation coefficients by using only the output images from cameras. The proposed algorithm is schematically illustrated in Figure 3.1. Our algorithm estimates the color interpolation coefficients in each local region through texture classification and linear approximation, and finds the CFA pattern that minimizes the interpolation errors [125, 128].

More specifically, we establish a search space of CFA patterns based on common

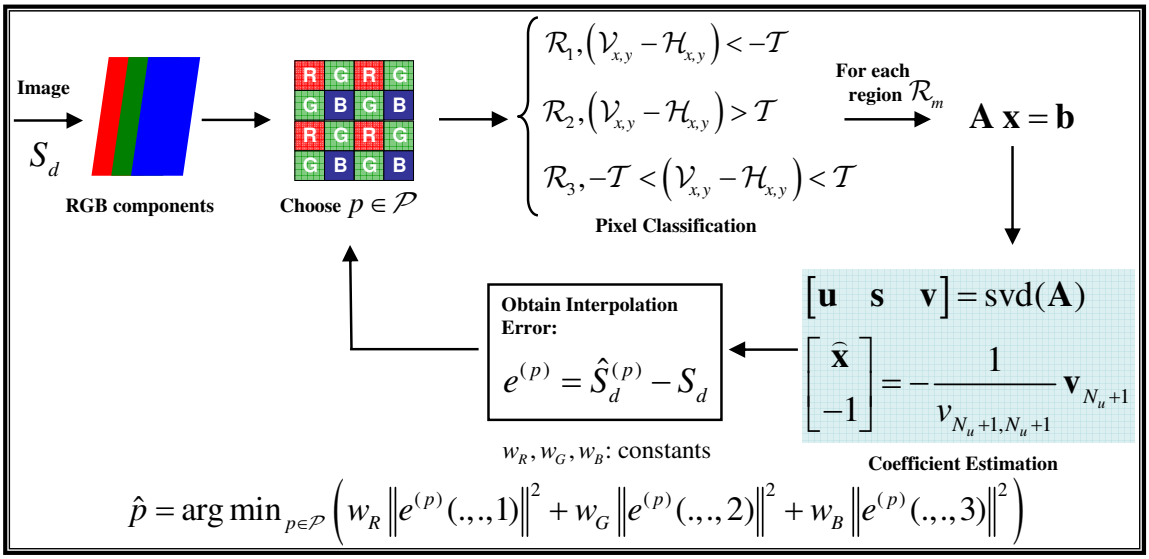


Figure 3.1: Algorithm to estimate color filter array and color interpolation coefficients.

practice in digital camera design. We observe that most commercial cameras use a RGB type of CFA with a fixed periodicity of 2×2 that can be represented as

$$\begin{array}{cc|c} C_1 & C_2 & \dots \\ C_3 & C_4 & \dots \\ \hline \vdots & \vdots & \ddots \end{array}$$

where $C_i \in \{R, G, B\}$ is the color of the corresponding sensor at a particular pixel location. In typical digital cameras, each of the three types of color sensors (R, G, and B) appears at least once in a 2×2 cell, resulting in a total of 36 possible patterns in the search space, denoted by $\mathcal{P}$. For every CFA pattern p in the search space $\mathcal{P}$, we estimate the interpolation coefficients in different types of texture regions of the image by fitting linear filtering models. These coefficients are then used to re-estimate the output image $\hat{S}_d^{(p)}$, and find the interpolation error $(\hat{S}_d^{(p)} - S_d)$. We now present the details of the proposed algorithm.

3.2.1 Texture Classification and Linear Approximation

We approximate the color interpolation to be linear in chosen regions of the image [123]. We divide the image into three kinds of regions based on the gradient features in a local neighborhood. Defining $I_{x,y} = S_d(x, y, p(x, y))$, the horizontal and vertical gradients at the location (x, y) can be found from the second order gradient values using

$$H_{x,y} = |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|, \tag{3.1}$$

$$V_{x,y} = |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|. \tag{3.2}$$

The image pixel at location (x, y) is classified into one of the three categories:

- *Region* $\Re_1$ contains those parts of the image with a significant horizontal gradient for which $(H_{x,y} - V_{x,y}) > T$, where T is a suitably chosen threshold;
- *Region* $\Re_2$ contains those parts of the image with a significant vertical gradient and is defined by the set of points for which $(V_{x,y} - H_{x,y}) > T$; and
- *Region* $\Re_3$ consists of the remaining parts of the image which are mostly smooth.

Using the final camera output S_d and the assumed sample pattern p, we identify the set of locations in each color of S_d that are acquired directly from the sensor array. We approximate the remaining pixels to be interpolated with a set of linear equations in terms of the colors of the pixels captured directly. In this process, we obtain nine sets of linear equations corresponding to the three types of regions $\Re_m (m = 1, 2, 3)$ and three color channels (R, G, B) of the image.

Let the set of N_e equations with N_u unknowns for a particular region and color channel be represented as $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ of dimension $N_e \times N_u$ and $\mathbf{b}$ of dimension $N_e \times 1$ specify the values of the pixels captured directly and those interpolated,

respectively, and $\mathbf{x}$ of dimension $N_u \times 1$ stands for the interpolation coefficients to be estimated. To cope with possible noisy pixel values in $\mathbf{A}$ and $\mathbf{b}$ due to other in-camera operations following interpolation (such as JPEG compression), we employ singular value decomposition [137] to estimate the interpolation coefficients. Let $\mathbf{A}_0$ and $\mathbf{b}_0$ represent the ideal values of $\mathbf{A}$ and $\mathbf{b}$ in the absence of noise, and the errors in $\mathbf{A}$ and $\mathbf{b}$ be denoted by $\mathbf{E}$ and $\mathbf{r}$, respectively, so that

$$\mathbf{A} = \mathbf{A}_0 - \mathbf{E}, \qquad \mathbf{b} = \mathbf{b}_0 - \mathbf{r},$$

The values of $\mathbf{x}$ are found by solving the minimization problem

$$\min_{\mathbf{E},\mathbf{r}} |||[\mathbf{E} \quad \mathbf{r}]|||_F,$$

subject to the constraint that $\mathbf{A}_0\mathbf{x} - \mathbf{b}_0 = 0$. Equivalently this can be written as

$$[\mathbf{A} + \mathbf{E}, \;\; \mathbf{b} + \mathbf{r}] \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = 0. \tag{3.3}$$

Here $||.||_F$ denotes the *Frobenius* norm of the matrix, so that

$$|||[\mathbf{E} \quad \mathbf{r}]|||_F = \left(\sum_{m=1}^{N_e} \sum_{n=1}^{N_u} |e(m,n)|^2 + \sum_{m=1}^{N_e} |r(m)|^2 \right)^{1/2}. \tag{3.4}$$

The solution to the minimization problem can be written as

$$\begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = -\frac{1}{v_{N_u+1,N_u+1}} \mathbf{v}_{N_u+1}, \tag{3.5}$$

where $\mathbf{v}_{N_u+1}$ represents the $(N_u+1)^{\text{th}}$ right singular vector of the combined matrix $[\mathbf{A} \;\; \mathbf{b}]$.

3.2.2 Finding the Interpolation Error and the CFA Sampling Pattern

Once we find the interpolation coefficients in each region, we use them to re-interpolate the sampled CFA output in the corresponding regions $\Re_m$, to obtain an estimate of the final output image $\hat{S}_d^{(p)}$. Here, the superscript p denotes that the output estimate is based on the choice of the CFA pattern p. The pixel-wise difference between the estimated final output and the actual camera output image is $e^{(p)} = \hat{S}_d^{(p)} - S_d$. The interpolation error matrix $e^{(p)}$ of dimension $H \times W \times C$ is obtained for all candidate search patterns $p \in \mathcal{P}$. Denoting the interpolation error in the red color component as $e^{(p)}(.,.,1)$ and so on, the final error is computed by a weighted sum of the errors of the three color channels:

$$\varepsilon(p) = w_R \, ||e^{(p)}(.,.,1)||_F^2 + w_G \, ||e^{(p)}(.,.,2)||_F^2 + w_B \, ||e^{(p)}(.,.,3)||_F^2 \tag{3.6}$$

The CFA pattern $\hat{p} = \arg\min_{p \in \mathcal{P}} \varepsilon(p)$ that gives the lowest overall absolute value of the weighted error is chosen as the estimated pattern. The constants w_R, w_G, and w_B denote the corresponding weights used for the three color components (red, green, and blue), and their values are based on the relative significance of the magnitude of errors in the three colors. In our experiments, we choose $w_R = w_B = 1$ and $w_G = 2$ to give more importance to the error in the green channel as it provides more information about the luminance values of the pixel [6]. The interpolation coefficients corresponding to the estimated CFA pattern $\hat{p}$ for all three types of regions and the three color channels are also obtained in this process. These coefficients can then be directly used to obtain the parameters of the components in the imaging model, as will be shown later in Section 3.3.2. They can also be processed to obtain further forensic evidence, as will be demonstrated by several

case studies in Section 3.4.

3.2.3 Reducing the Search Space for CFA Patterns

The search space for the CFA patterns can be reduced using a hierarchial approach. As an example, we synthetically generate a 512×512 image, sample it on the Bayer pattern, and interpolate using the bicubic method. In Figure 3.2, we show the detection statistics $d_s(p)$ given by

$$d_s(p) = \frac{\varepsilon(p)}{H \times W \times (w_R + w_G + w_B)}, \tag{3.7}$$

and sorted in ascending order for the 36 different CFA patterns. In this case, the Bayer pattern gave the lowest interpolation error and was correctly identified. A closer look at the results in Figure 3.2 reveals that the detection statistics form three separate clusters, with some values close to 0, some around $0.3 - 0.4$, and others close to 0.7. A similar trend is also observed for real camera data and other synthetically generated images sampled on different CFA patterns and interpolated with the six representative interpolation techniques reviewed in Appendix I of this chapter. This observation forms the basis for the heuristic discussed in this subsection to reduce the search space of the CFA patterns.

Figure 3.3 shows sample patterns from these three clusters. *Cluster 1* includes all 2×2 patterns that have the same color along diagonal directions (either along the main diagonal or off-diagonal), chosen among the three colors (red, green, or blue). The remaining two spots can be filled in two different ways, giving a total of 12 such patterns in the first cluster. *Cluster 2* and *Cluster 3* consists of patterns that have the same color along the horizontally (or vertically) adjacent blocks of the 2×2 grid. *Cluster 2* has either red or blue color repeated to produce a total of 16 possible patterns. The remaining eight patterns with green appearing twice

form *Cluster 3*. In this example, the Bayer pattern is the actual color filter array and the patterns from first cluster give lower errors compared to the other clusters. The patterns from *Cluster 3* gives the highest error values because the error in the green color channel is penalized more with the weight assignment $w_G = 2$ and $w_R = w_B = 1$ in (3.6).

The observation of clustering of patterns into three groups helps us develop a heuristic to reduce the search space of CFA patterns. We first divide the 36 patterns into three groups and choose one representative pattern from each of the three classes. The interpolation error is then estimated for these representative patterns to find the cluster that the actual CFA pattern is most likely to belong. Finally, a full search is performed on the chosen cluster to find the pattern with the lowest interpolation error. The number of searches required to find the optimal solution can be reduced to around 10. If additional information about the patterns are available, it may be used to further reduce the search space. For instance, a forensic analyst may choose to test only on those CFA patterns that have two green color components if he/she has such prior knowledge about the visual sensor.

3.2.4 Evaluating Confidence in Component Parameter Estimation

In addition to identifying the parameters of the internal building blocks of the camera, it is also important to know the confidence level on the estimation result. A higher confidence value in estimation would increase the trustworthiness of the decision made by a forensic analyst.

We propose an entropy based metric to quantify the confidence level on the estimation result. Given a test image, we estimate its interpolation coefficients and

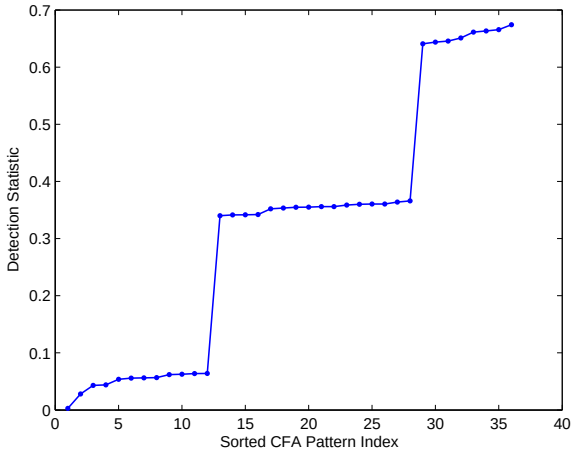


Figure 3.2: Sorted detection statistics in terms of normalized overall error for different candidate search patterns.

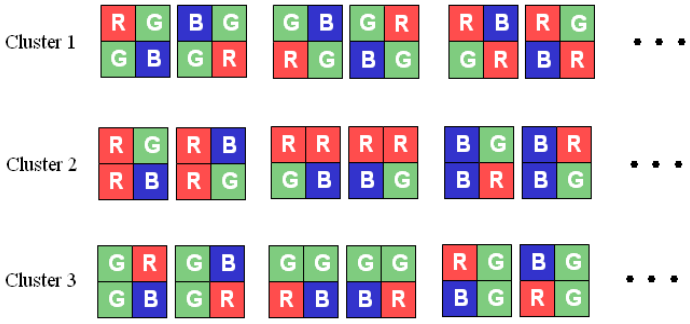


Figure 3.3: Sample CFA patterns from the three clusters.

provide it as an input to a $c-$class SVM classifier that is trained on the coefficients of the c candidate interpolation methods. The probability that a given test sample comes from the i^{th} class, q_i, is estimated from the soft decision values using the probabilistic SVM framework [148], and the test data point is classified into class k if q_k is larger than the other probabilities. Some details of the probabilistic SVMs are included in Appendix II of this chapter for readers' reference. The confidence

score η on the decision is then defined as

$$\eta = 2\Upsilon\left(1 - \frac{\sum_{i=1}^{c} q_i \log_2\left(\frac{1}{q_i}\right)}{\log_2 c}\right). \tag{3.8}$$

where $\Upsilon(y) = z$ is defined as the inverse binary entropy function such that

$$y = -z\log_2(z) - (1-z)\log_2(1-z) \text{ for } 0 \le z \le \frac{1}{2}.$$

The argument to the Υ function in (3.8) measures the entropy difference between the distribution $\{q_i\}$ and a discrete uniform distribution, and the final value of η is normalized to the range of [0, 1] to represent a probability.

To verify that the proposed metric η can reflect the confidence level, we examine two extreme cases. When $\mathbf{q} = [1,\ 0,\ 0,\ \ldots, 0]$, the decision of choosing the first class is made with a very high confidence and $\eta = 1$. And when $\mathbf{q} = [\frac{1}{c} + \epsilon,\ \frac{1}{c} - \frac{\epsilon}{c-1},\ \frac{1}{c} - \frac{\epsilon}{c-1},\ \ldots, \frac{1}{c} - \frac{\epsilon}{c-1}]$ where ϵ is a small positive real number, there is an almost equal probability that the given data sample comes from any of the c classes. In this case, the decision is made with a very low confidence and η also approaches zero. For other values of $\mathbf{q}$ between these two extreme cases, the value of η would lie in the interval [0, 1], with a higher value indicating more confidence in the decision.

3.3 Experimental Results

3.3.1 Simulation Results with Synthetic Data

We use synthetic data constructed from 20 representative images to study the performance of the proposed techniques. The original images are first downsampled to remove the effect of previously applied filtering and interpolation operations.

They are then sampled on the three different CFA patterns as shown in Figure 2.2. Each of the sampled images are interpolated using one of the six interpolation methods reviewed in Appendix I of this chapter, namely, (a) Bilinear, (b) Bicubic, (c) Smooth Hue, (d) Median Filter, (e) Gradient based, and (f) Adaptive Color Plane. Thus, our total dataset contains $20 \times 3 \times 6 = 360$ images, each of size 512×512.

Simulation Results under no Post-processing

We test the proposed CFA pattern and color interpolation identification algorithms on this synthetic data set. In the noiseless case with no post-processing, we observe no errors in estimating the CFA pattern. We use a 7×7 neighborhood to estimate the interpolation coefficients for the three color components in the three types of texture regions, and pass it to a classifier to identify the interpolation algorithm. A support vector machine (SVM) classifier with a third-degree polynomial kernel [19] [22] is used to identify the interpolation method. We randomly choose 8 out of the 20 images from each of the six interpolation techniques as ground truth for training and the remaining 12 images for testing. We repeat the experiment 500 times with a random set of images each time. The classifier is 100% accurate in identifying the correct color interpolation algorithm without any errors.

Simulation Results with Post-processing

As mentioned earlier, post-processing such as color correction and compression are commonly done in nearly all commercial cameras. Therefore, to derive useful forensic evidence from output images, it is very important that the proposed methods be robust to the common post-processing operations done in cameras.

In this work, we primarily focus on JPEG compression and additive noise, and study the performance under these distortions. Other post-processing operations such as color correction and white-balancing are typically multiplicative, where the final image is obtained by multiplying the color interpolated image by appropriately chosen constants in the camera color space. In most commercial cameras, white balancing is done in the XYZ color space [150], and the inverse transformation may be applied before estimating the color interpolation coefficients. The multiplicative factors used in white balancing operations operate on each color channel separately [39], and therefore white balancing operations do not significantly affect our solution of the color interpolation coefficients. Gamma correction can be estimated from the final output images [34] and can be undone before computing the interpolation coefficients. For the results presented in this sub-section, we directly obtain the coefficients from the output images and do not perform inverse gamma correction based on the estimated values of gamma. Later in Section 3.3.2, we show that the estimation results are robust to gamma correction distortions.

(*i*) Performance Results Under JPEG compression: JPEG compression is an important post-processing operation that is commonly done in cameras. The noise introduced by compression could potentially result in errors in estimating the color interpolation coefficients and the CFA pattern. We test the proposed CFA pattern identification algorithm with the synthetic data obtained under different JPEG quality factors $\{20, 30, \ldots, 80, 90, 99\}$. We find that in all cases, the estimator gives very good results and the correct CFA pattern is always identified.

Next, we study the accuracy in identifying the color interpolation when the synthetically generated images are JPEG compressed. Here, we consider two possible scenarios. In the first case, a forensic analyst does not have access to the

camera(s) and therefore does not have control over the input(s) to the device. He/she makes a judgement based on the forensic evidence obtained from the images submitted for trial. In this scenario, the pictures obtained with different interpolation methods would correspond to different scenes, which we shall call as the *multiple-scene* case. The performance of the proposed color interpolation identification for the multiple-scene case at different JPEG quality factors is shown in Figure 3.4(a). Here we use a total of 12 images (two distinct images for each of the six interpolation methods) for training, and test with the remaining 8 images under each interpolation ($8 \times 6 = 48$ in total). The experiment is repeated 500 times by choosing a random training set each time. We observe that the average percentage of images for which the interpolation technique is correctly identified is around 95–100% for moderate to high JPEG quality factors of 80–100[1] and the average performance reduces to 80–85% for quality factors from 50–80.

Alternatively, if a forensic analyst has access to the camera, he/she can perform controlled testing by choosing the input to the cameras so as to reduce the impact of the input's variation on the forensic analysis. In this scenario, the analyst may consider taking similar images with all the cameras under study, in order to improve the estimation accuracy and increase the confidence level on his/her final judgement. We call this situation the *single-scene* case. The single-scene case corresponds to the *semi non-intrusive* forensic analysis discussed earlier in Chapter 1. The performance of the proposed color interpolation technique for this case for different JPEG quality factors is shown in Figure 3.4(b). Here we use 8 images under the six interpolation techniques for training (48 in total) and the

[1]Most commercial digital cameras employ JPEG compression with quality factors between 80 and 100

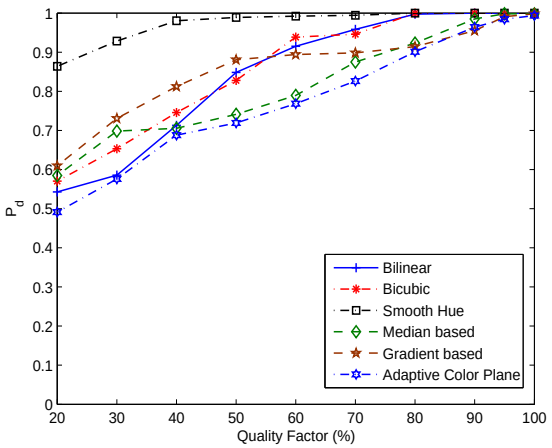
72 remaining images for testing. We observe that for most JPEG quality factors, the average percentage of images for which the color interpolation technique is correctly identified is around 96% and thus the forensic decision can be made with a higher confidence compared to the multiple-scene case. The accuracy can be further improved using more images with representative characteristics for training. This suggests that with an increasing number of well-designed image inputs to the system, the detection performance can be enhanced.

***(ii)* Performance Results Under Additive Noise:** Additive noise can be used to model the sensor noise and several other kinds of random post-processing operations that may occur during the scene capture process. In order to study the noise resilience of a forensics system, we test the proposed CFA pattern identification algorithm with the images obtained under different noise levels with peak-signal to noise ratios (PSNRs) of $15, 20, 30$, and 40 dB, respectively. The correct CFA pattern was identified in all but one cases, and the only error occurred at an extremely low PSNR of 15dB for an image interpolated with the adaptive color plane method. Even in this case, the correct pattern came in the top three results.

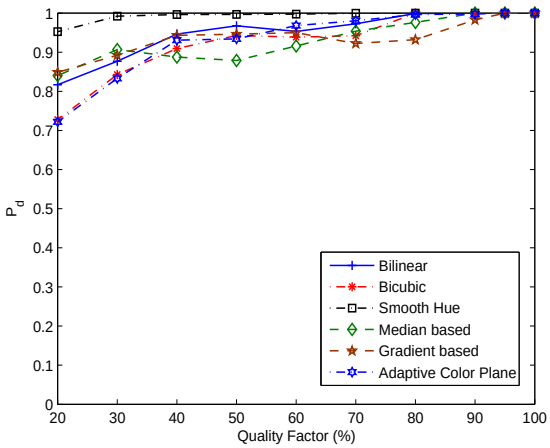
We then study the identification performance of the color interpolation method under additive noise. The performance for synthetic data, averaged over 500 iterations, for the multiple-scene and the single-scene case are shown in Figure 3.5(a) and Figure 3.5(b), respectively. We observe that there is around 90% accuracy for the multiple-scene case and it increases to around 95% for the single-scene scenario.

3.3.2 Results on Camera Data

A total of 19 camera models as shown in Table 3.1 are included in our experiments. For each of the 19 camera models, we have collected about 40 images. The images



(a) Multiple-Scene case

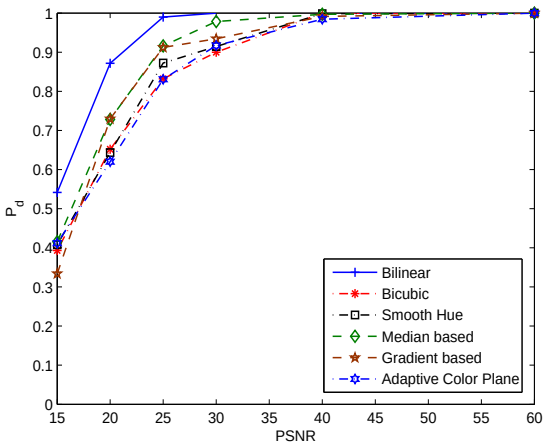


(b) Single-Scene case

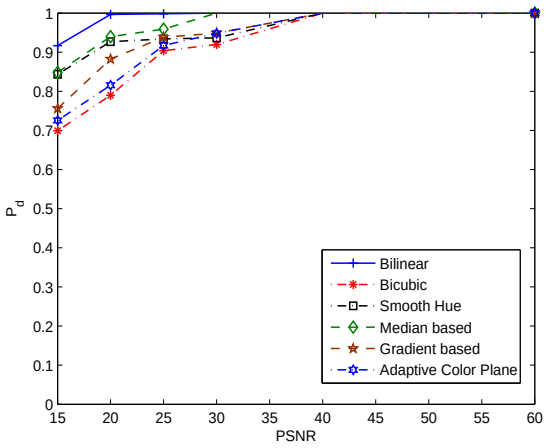
Figure 3.4: Fraction of images for which the color interpolation technique is correctly identified under different JPEG compression quality factors. The testing results here are with the synthetic dataset.

from different camera models are captured under uncontrolled conditions−different sceneries, different lighting situations, and compressed under different JPEG quality factors as specified by default values in each camera. The default camera settings (including image size, color correction, auto white balancing, JPEG compression, etc.) are used in image acquisition. From each of these images, we randomly choose five non-overlapping 512×512 blocks per image and use it for subsequent analysis. Thus, our database consists of a total of 3800 different 512×512 pictures with 200 samples for each of the 19 camera models.

Note that all the cameras in our database use RGB type of CFA pattern with red, green, and blue sensors. The search space for CFA in our experiments focusses on such RGB type CFA, since it has been widely employed in digital camera design and most cameras in the market currently use this pattern or its variations. There are a few exceptions in CFA designs, for example, some models use CMYG type of



(a) Multiple-Scene case



(b) Single-Scene case

Figure 3.5: Fraction of images for which the color interpolation technique is correctly identified under different noise PSNR's. The testing results here are with the synthetic dataset.

CFA that captures the cyan, magenta, yellow, and green components of light [7]. We believe that the proposed algorithms may be extended to identify CMYG type CFA patterns by incorporating an appropriate set of CMYG combinations in the search space, and we plan to test cameras with such patterns as part of our future work.

Among RGB type CFA patterns, several layouts of the three types of color filters have been used in practice. The 2×2 square arrangement is the most popular and most digital cameras utilize a shifted variation of the Bayer pattern to capture the real world scene. Recently introduced super CCD cameras [3] have sensors placed as shown in Figure 3.6. To test the performance of the proposed algorithms to such cameras, we include images from the Fujifilm Finepix A500 (camera no. 17) that uses super CCD [3] in our database.

As an initial step, we try to estimate the CFA pattern from the output images

Table 3.1: Camera models used in experiments.

No.	Camera Model	No.	Camera Model
1	Canon Powershot A75	11	Olympus C3100Z/C3020Z
2	Canon Powershot S400	12	Olympus C765UZ
3	Canon Powershot S410	13	Minolta DiMage S304
4	Canon Powershot S1 IS	14	Minolta DiMage F100
5	Canon Powershot G6	15	Casio QV 2000UX
6	Canon EOS Digital Rebel	16	FujiFilm Finepix S3000
7	Nikon E4300	17	FujiFilm Finepix A500
8	Nikon E5400	18	Kodak CX6330
9	Sony Cybershot DSC P7	19	Epson PhotoPC 650
10	Sony Cybershot DSC P72		

Figure 3.6: Super CCD sensor pattern.

using the algorithm described in Section 3.2. The estimation results show with a high confidence that all the cameras except Fujifilm Finepix A500 (camera no. 17) use shifted versions of the Bayer color filter array as their CFA pattern. For instance, the estimated 2×2 CFA that minimized the fitting errors on JPEG images from Canon EOS Digital Rebel (camera no. 6) and the Fujifilm Finepix S3000 (camera no. 16) are shown in Figure 3.7(a) and (b), respectively. The estimation results perfectly match these cameras' ground-truth data obtained by reading the headers of the raw image files produced by the two cameras.

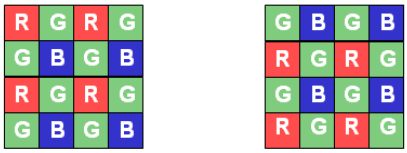


Figure 3.7: Sample CFA patterns for (a) Canon EOS Digital Rebel and (b) Fujifilm Finepix S3000.

When testing the images from the Fujifilm Finepix A500 (camera no. 17) with the same 36 square patterns in the CFA pattern search space, we notice that the best 2×2 pattern in the search space is still a shifted version of the Bayer pattern. However, we observe that the minimum error ε, as given by (3.6), is larger than the ones obtained from other square-CFA cameras. Therefore, the overall decision confidence is lower for this super CCD camera compared to the other cameras in the database. Further, we also find that the CFA pattern estimation results are not consistent across different images taken with the same camera, i.e., different images from Fujifilm Finepix A500 give different shifted versions of the Bayer pattern as the estimated CFA. Such inconsistencies in the results along with lower confidence in parameter estimation could be an indication that the camera does not employ a square CFA pattern. One possible approach to identify super CCD is to enlarge the CFA search space to include these patterns. We plan to further investigate this aspect in our future work to gather forensic evidence to distinguish super CCD cameras and square CFA cameras.

Next, we try to estimate the color interpolation coefficients in different image regions using the algorithm presented in Section 3.2.2. In our simulations, we find the coefficients of a 7×7 filter in each type of region and color channel, thus giving a total of $7 \times 7 \times 3 \times 3 = 441$ coefficients per image. Sample coefficients obtained

0	0.007	0	0.027	0	0.014	0
0.008	0	-0.062	0	-0.070	0	0.003
0	-0.041	0	0.435	0	-0.032	0
-0.004	0	0.218	1.000	0.204	0	0.001
0	-0.034	0	0.441	0	-0.038	0
0.005	0	-0.075	0	-0.064	0	0.002
0	0.013	0	0.030	0	0.012	0

(a)

0	0.006	0	0.009	0	0.009	0
0.005	0	-0.070	0	-0.069	0	0.007
0	-0.023	0	0.213	0	-0.019	0
0.019	0	0.414	1.000	0.399	0	0.029
0	-0.014	0	0.214	0	-0.020	0
0.002	0	-0.074	0	-0.064	0	0.006
0	0.008	0	0.011	0	0.005	0

(b)

0	-0.005	0	0.008	0	0.001	0
0.018	0	-0.079	0	-0.050	0	0.006
0	-0.021	0	0.315	0	-0.023	0
0.003	0	0.325	1.000	0.301	0	0.023
0	-0.023	0	0.301	0	-0.015	0
-0.006	0	-0.061	0	-0.051	0	0.010
0	0.002	0	0.018	0	-0.001	0

(c)

0	0	0	0.004	0	0	0
0	0	-0.035	0	-0.035	0	0
0	-0.035	0	0.316	0	-0.035	0
0.004	0	0.316	1.000	0.316	0	0.004
0	-0.035	0	0.316	0	-0.035	0
0	0	-0.035	0	-0.035	0	0
0	0	0	0.004	0	0	0

(d)

Figure 3.8: Interpolation coefficients for the green channel for one sample image taken with the Canon Powershot A75 camera for (a) Region $\Re_1$ with significant horizontal gradient, (b) Region $\Re_2$ with significant vertical gradient, (c) Smooth region $\Re_3$, (d) Coefficients of bicubic interpolation.

using the Canon Powershot A75 camera for the three types of regions in the green image are shown in Figure 3.8. For region $\Re_1$ that corresponds to areas having significant horizontal gradient, we observe that the value of the coefficients in the vertical direction (0.435 and 0.441) are significantly higher than those in the horizontal directions (0.218 and 0.204). This indicates that the interpolation is done along the edge, which in this case is oriented along the vertical direction. Similar corresponding inferences can be made from coefficients in region $\Re_2$ of significant vertical gradient. Compared to these two regions, the coefficients in region $\Re_3$ have

almost equal values in all four directions, and do not have any directional properties. Moreover, careful observation of the coefficients in region $\Re_3$ reveals their close resemblance to the bicubic interpolation coefficients shown in Figure 3.8(d). This suggests that it is very likely that the Canon Powershot A75 camera uses bicubic interpolation for smooth regions of the image. Similar results obtained for other camera models indicate with $\eta = 96\%$ confidence that all cameras use the bicubic interpolation for handling smooth regions. This is consistent with common knowledge in image processing practice that bicubic interpolation is good for regions with slowly changing intensity values [63].

3.4 Case Studies and Applications of Non-Intrusive Forensic Analysis

In this section, we present case studies to illustrate the applications of the proposed non-intrusive forensic analysis methodology for camera identification (acquisition forensics), and for providing clues to identify infringement/licensing.

3.4.1 Identifying Camera Brand from Output Images

The color interpolation coefficients estimated from the image can be used as features to identify the camera brand utilized to capture the digital image. As shown in Section 3.3.2, most cameras employ similar kinds of interpolation techniques for smooth regions. Therefore, we focus on non-smooth regions and use the coefficients obtained from the horizontal gradient regions $\Re_1$ and vertical gradient regions $\Re_2$ as features to construct a camera brand identifier.

To obtain more reliable forensic evidence from the input image for camera

identification, we first pre-process the image by edge detection to locate five significant 512×512 blocks with the highest absolute sum of gradient values. The interpolation coefficients corresponding to the regions $\Re_1$ and $\Re_2$, from all three color channels, estimated from these 512×512 blocks are used as features for identification.

We use a classification based framework to identify camera brand. For each camera in the database, we collect 40 different images and obtain 200 different 512×512 image blocks by locating the top five regions with higher gradient values. These 200 image blocks collected from each of the 19 cameras are grouped so that all images from the same brand form one class. A $9-$camera brand SVM classifier with a polynomial kernel function [22] is constructed with 50% of the images randomly chosen from each class for training. The remaining images are used in testing and the process is repeated 500 times by randomly choosing a training set each time. Table 3.2 shows the average confusion matrix, where the $(i, j)^{\text{th}}$ element gives the percentage of images from camera brand$-i$ that are classified to belong to camera brand$-j$. The main diagonal elements represent the classification accuracy and achieve a high average classification rate of 90% for nine camera brands. A closer look at the remaining 10% of misclassified images suggest that most of them have significant amount of smooth regions; these regions have less discriminating capability because most digital cameras employ similar kind of interpolation in the smooth regions as demonstrated earlier.

The above results demonstrate the effectiveness of using the color interpolation component as features to differentiate different camera brands. The robustness of estimating these features under JPEG and additive noise has been shown earlier in Section 3.3.1. Here we further examine the robustness against such nonlinear

Table 3.2: Confusion matrix for identifying different camera brands (* denotes values smaller than 4%).

	Canon	Nikon	Sony	Olympus	Minolta	Casio	Fuji	Kodak	Epson
Canon	96%	*	*	*	*	*	*	*	*
Nikon	*	83%	5%	*	*	*	*	*	*
Sony	*	*	90%	*	*	*	*	*	*
Olympus	*	*	*	93%	*	*	*	*	*
Minolta	8%	*	*	*	81%	*	*	*	*
Casio	*	*	*	6%	*	89%	*	*	*
Fuji	*	*	*	*	7%	*	87%	*	*
Kodak	*	*	*	*	*	*	*	89%	*
Epson	*	*	*	*	*	*	*	*	100%

point operations as gamma correction. As a common practice in digital camera design, most cameras perform gamma correction with a $\gamma = 1/2.2$ to match the luminance of the digital image with that of the display monitor. In order to test the goodness of the proposed algorithms for gamma correction, we first do inverse gamma correction with $\gamma = 2.2$ on the original camera images.[2] The interpolation coefficients are then estimated from these gamma corrected images and used in camera brand identification. In this case, the confusion matrices are similar to the ones in Table 3.2, and average identification accuracy was estimated to be 89%. This negligible difference from the non-gamma correction case of 90% suggests that the camera identification results are invariant to gamma correction in digital

[2]In a general scenario, the value of γ can be estimated from the output images [34] and the corresponding inverse could be applied before estimating the interpolation coefficients.

cameras.

As the problem of camera brand identification only received attention recently, there is a very limited amount of related work to compare with. Some algorithms were developed recently in [70] [15], where the authors test their algorithms for pictures taken under controlled conditions with the same scene captured with multiple cameras (corresponding to the *single-scene* case discussed earlier in Section V-A). The best performance initially reported in [15] is 84% on three brands, and this algorithm is sensitive to other in-camera processing such as compression owing to the dependence on image content by the null-based spectral features employed in [15]. Concurrent to the present work, further improvements have been made to the algorithm in [15] by separately obtaining the coefficients from smooth and non-smooth regions of each image, leading to an enhanced classification accuracy of 96% for three camera brands [14]. Compared to these alternative approaches, the interpolation coefficients derived in our work by exploring the spatial filtering relations are less dependent on input scenes and are robust against various common in-camera processing. The formulation of minimizing noise norm via (3.5) further helps mitigate the impact from noise, compression, and other in-camera processing. As a result, the features obtained from the proposed component forensics methodologies are able to achieve a high classification accuracy over a much larger dataset with 19 camera models from nine different brands. Further, as will be demonstrated later in this section, the proposed component forensic techniques have a broader goal of identifying the algorithms and parameters employed in various components in digital cameras, and are not restricted to camera brand identification.

3.4.2 Identifying Camera Model from Output Images

Our results in the previous subsection demonstrate the robustness of non-intrusively identifying the camera brand using the color interpolation coefficients as features. In this subsection, we extend our studies to answer further forensic questions to find the exact camera model used to capture a given digital image, and examine the performance in identifying the camera model.

We use 200 images from each of the 19 cameras in our experiments. Out of these 200 images, a randomly chosen 125 images are used for training and the remaining are for testing with a $19-$camera model SVM classifier. The simulation is repeated 500 times with different training sets and the average confusion matrix is shown in Table 3.3. The $(i,j)^{\text{th}}$ element in the confusion matrix gives the fraction of images from camera model$-i$ classified as camera model$-j$. In order to highlight the significant values of the table, we show only those set of values that are greater than or equal to a chosen threshold $\lambda = 1/N_c$, where N_c is the number of cameras ($\lambda = 1/19$ in our experiments). The average classification accuracy is 86% for 19 camera models.

The classification results reveal some similarity among different camera models in handling interpolation, as there are some off-diagonal elements that have a non-zero value greater than the threshold of 1/19. For example, among the Canon Powershot S410 (camera no. 3) images, 20% were classified as belonging to Canon Powershot S400 (camera no. 2). A similar trend is also observed for images from other Canon models. These results indicate that the color interpolation coefficients are quite similar among the Canon models and hence it is likely that they are using similar kinds of interpolation methods.

Table 3.3: Confusion matrix for identifying different camera models. The matrix is divided based on different camera makes. The values below the threshold $\lambda = \frac{1}{19}$ are denoted by $*$. The camera index numbers are according to Table 3.1.

		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Canon	01	0.92	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	02	*	0.84	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	03	*	0.20	0.80	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	04	*	0.16	*	0.56	*	*	*	*	0.16	*	*	*	*	0.08	*	*	*	*	*
	05	*	*	0.08	*	0.88	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	06	*	*	*	0.08	*	0.88	*	*	*	*	*	*	*	*	*	*	*	*	*
Nikon	07	*	*	*	*	0.16	*	0.60	*	*	0.08	*	*	*	0.08	*	*	*	*	*
	08	*	*	*	*	*	*	*	0.72	*	*	*	*	*	0.08	*	*	0.16	*	*
Sony	09	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*	*	*
	10	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*	*
Olympus	11	*	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*
	12	*	*	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*
Minolta	13	*	*	*	*	*	*	*	*	*	*	*	*	0.96	*	*	*	*	*	*
	14	0.08	*	*	*	*	*	*	*	*	*	*	0.08	*	0.80	*	*	*	*	*
Casio	15	*	0.16	*	*	*	*	*	*	*	*	*	*	*	*	0.84	*	*	*	*
Fujifilm	16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0.96	*	*	*
	17	*	*	0.08	*	*	*	*	*	*	*	*	*	*	0.20	*	*	0.68	*	*
Kodak	18	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0.08	*	*	0.88	*
Epson	19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1.00

3.4.3 Similarities in Camera Color Interpolation Algorithms

Motivated by the results in the previous subsection, we further analyze the similarity between the camera models in this subsection, and propose metrics to quantitatively evaluate the closeness among interpolation coefficients from several cameras.

Studying Similarities in Cameras using Leave-One-Out

We perform additional experiments to identify the camera models with similar color interpolation by a *leave-one-out* procedure. More specifically, we train the classifier by omitting the data from one of the camera models and test it with these coefficients, to find the nearest neighbor in the color interpolation coefficient space. For instance, when we train the SVM using all the 200 images from 18 cameras except Canon Powershot S410 (Camera no. 3), and then test it using the 200 images from Canon Powershot S410, we observe that 66% of the Canon

Powershot S410 images are classified as Canon Powershot S400. Furthermore, out of the remaining images, 28% of the pictures are classified as one of the remaining Canon models. The reverse trend is also observed when we train with all the images except Canon Powershot S400 (camera no. 2) and use these images for testing. Around 45% of the Canon Powershot S400 pictures are classified as Canon Powershot S410, 19% are categorized as Canon Powershot A75, and 15% of the remaining guessed as some other Canon model. This result suggests that there is a considerable amount of similarity in the kind of interpolation algorithms used by various Canon models.

A similar trend is also observed for the two Sony cameras in our database. We note that around 66% of the Sony Cybershot DSC P7 model are classified as Sony Cybershot DSC P72 model when the former was not used in training. These results indicate the similarities in the kind of interpolation algorithm among various models of the same brand. Interestingly, we also observe similarity between Minolta DiMage S304 and Nikon E4300. Around 53% of the Minolta DiMage S304 pictures are designated as Nikon E4300 camera model. This suggests closeness between the interpolation coefficients in the feature space.

Quantifying Similarity in Color Interpolation with a Divergence Score

From our preliminary analysis in Section 3.3.2, we observe that the majority of the cameras use similar kinds of interpolation techniques in handling smooth regions. We thus focus our attention on the type of interpolation used by a camera in the non-smooth regions. We extend our interpolation coefficient estimation model in Section 3.2.2 to explicitly target at non-smooth regions in the image. To do so, we divide the image into *eight* types of regions depending on the relative gradient esti-

mates in eight directions (namely north, east, west, south, north-east, north-west, south-east, and south-west). The gradient values can be obtained following the threshold-based variable number of gradients (VNG) algorithm [23]. For example, the gradient in the north direction J_N is obtained using

$$\begin{aligned} J_N(x,y) &= |I_{x-1,y} - I_{x+1,y}| + |I_{x-2,y} - I_{x,y}| \\ &+ 0.5 \times |I_{x-1,y-1} - I_{x+1,y-1}| + 0.5 \times |I_{x-1,y+1} - I_{x+1,y+1}| \\ &+ 0.5 \times |I_{x-2,y-1} - I_{x,y-1}| + 0.5 \times |I_{x-2,y+1} - I_{x,y+1}|, \end{aligned} \tag{3.9}$$

where $I_{x,y} = S_d(x, y, p(x, y))$ represents the image pixel sample. Similar expressions for gradients in the remaining seven directions can be developed to find the local gradient values [23]. Once these gradients are obtained, they are compared to a threshold to divide the image into eight types of texture regions. The interpolation coefficients are obtained in each region by solving a set of linear equations as given by (3.5).

We use a classification based methodology to study the similarities in interpolation algorithms used by different cameras. To construct classifiers, we start with 100 representative images, downsample them (by a factor of 2) and then re-interpolate with each of the six different interpolation methods as discussed in Section 3.3.1. With a total of 600 images synthetically generated in this way, we run the color interpolation estimator to find the coefficients for each image. The estimated coefficients are then used to train a 6-class SVM classifier, where each class represents one interpolation method. After training the SVM classifier, we use it to test the images taken by the 19 cameras. For each of the 200 images taken by every camera in the 19-camera dataset, we estimate the CFA parameters (eight sets of coefficients each with a dimension of 5×5), feed them as input to the above

classifier and record the classification results.[3] Probabilistic SVM framework is used in classification and the soft decision values are recorded for each image [148] (refer to Appendix II of this chapter for more details). If the two camera models employ different interpolation methods (not necessarily the same as the six typical methods in the classifier), then the classification results are likely to be quite different, and their differences can be quantified by an appropriate distance between the classification results.

More specifically, for each image in the database, the interpolation coefficients are found and fed into the N-class classifier, where N denotes the number of possible choices of the interpolation algorithms studied ($N = 6$ in our experiments). Let the output of the classifier be denoted as a probability vector $\underline{g} = [g_1, g_2, \ldots, g_N]$, where g_k gives the probability that the input image employs the interpolation algorithm$-k$ ($1 \leq k \leq N$). Such probability vectors are obtained for every image in the database and the average performance is computed for each camera model. Let the average classification results for camera model$-i$ be represented by the vector $\underline{\pi_i} = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{iN}]$, where π_{ik} is the average probability for an image from camera model$-i$ to be classified as using the interpolation algorithm$-k$. The π_{ik}'s are estimated using soft decision values obtained using the probabilistic SVM framework. The similarities of the interpolation algorithms used by any two cameras (with indices i and j) can now be measured in terms of a *divergence* score φ_{ij}, defined as symmetric *Kullback-Leibler* (KL) distance between the two probability

[3] A kernel size of 5×5 is chosen in this case to limit the total number of coefficients, and to make the total number of features to be on the same order of magnitude as the previous case in Section 3.4.2 where we used a kernel size of 7×7 and three gradient based regions.

Table 3.4: Divergence scores for different camera models as indexed in Table 3.1. The values below or equal to 0.06 are shaded, and the $*$ indicates zero similarities between the same camera models by definition.

		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Canon	01	*	0.06	0.06	0.17	0.14	0.35	0.36	0.68	0.22	0.76	0.25	0.31	0.18	0.09	0.31	0.18	0.54	0.83	0.25
	02	0.06	*	0.05	0.07	0.05	0.19	0.19	0.46	0.11	0.55	0.12	0.14	0.11	0.03	0.15	0.17	0.35	0.57	0.16
	03	0.06	0.05	*	0.15	0.06	0.18	0.22	0.47	0.22	0.51	0.23	0.27	0.20	0.10	0.29	0.33	0.31	0.57	0.27
	04	0.17	0.07	0.15	*	0.10	0.22	0.23	0.50	0.14	0.71	0.04	0.08	0.10	0.07	0.10	0.19	0.49	0.58	0.23
	05	0.14	0.05	0.06	0.10	*	0.07	0.14	0.36	0.14	0.46	0.19	0.14	0.16	0.09	0.16	0.32	0.25	0.39	0.26
	06	0.35	0.19	0.18	0.22	0.07	*	0.15	0.36	0.18	0.42	0.32	0.21	0.30	0.22	0.23	0.54	0.23	0.35	0.37
Nikon	07	0.36	0.19	0.22	0.23	0.14	0.15	*	0.21	0.19	0.21	0.24	0.16	0.12	0.18	0.17	0.47	0.12	0.19	0.34
	08	0.68	0.46	0.47	0.50	0.36	0.36	0.21	*	0.53	0.16	0.47	0.39	0.39	0.48	0.41	0.89	0.31	0.10	0.92
Sony	09	0.22	0.11	0.22	0.14	0.14	0.18	0.19	0.53	*	0.09	0.17	0.08	0.11	0.07	0.07	0.13	0.43	0.61	0.14
	10	0.76	0.55	0.51	0.71	0.46	0.42	0.21	0.16	0.09	*	0.66	0.61	0.52	0.56	0.59	1.02	0.18	0.23	0.82
Olympus	11	0.25	0.12	0.23	0.04	0.19	0.32	0.24	0.47	0.17	0.66	*	0.11	0.10	0.08	0.11	0.19	0.56	0.61	0.25
	12	0.31	0.14	0.27	0.08	0.14	0.21	0.16	0.39	0.08	0.61	0.11	*	0.08	0.12	0.01	0.20	0.42	0.42	0.26
Minolta	13	0.18	0.11	0.20	0.10	0.16	0.30	0.12	0.39	0.11	0.52	0.10	0.08	*	0.06	0.08	0.17	0.39	0.45	0.25
	14	0.09	0.03	0.10	0.07	0.09	0.22	0.18	0.48	0.07	0.56	0.08	0.12	0.06	*	0.11	0.11	0.42	0.61	0.13
Casio	15	0.31	0.15	0.29	0.10	0.16	0.23	0.17	0.41	0.07	0.59	0.11	0.01	0.08	0.11	*	0.18	0.44	0.45	0.24
Fujifilm	16	0.18	0.17	0.33	0.19	0.32	0.54	0.47	0.89	0.13	1.02	0.19	0.20	0.17	0.11	0.18	*	0.82	1.05	0.17
	17	0.54	0.35	0.31	0.49	0.25	0.23	0.12	0.31	0.43	0.18	0.56	0.42	0.39	0.42	0.44	0.82	*	0.23	0.51
Kodak	18	0.83	0.57	0.57	0.58	0.39	0.35	0.19	0.10	0.61	0.23	0.61	0.42	0.45	0.61	0.45	1.05	0.23	*	0.98
Epson	19	0.25	0.16	0.27	0.23	0.26	0.37	0.34	0.92	0.14	0.82	0.25	0.26	0.25	0.13	0.24	0.17	0.51	0.98	*

distributions $\underline{\pi_i}$ and $\underline{\pi_j}$:

$$\varphi_{ij} = D(\underline{\pi_i}||\underline{\pi_j}) + D(\underline{\pi_j}||\underline{\pi_i}), \tag{3.10}$$

$$\text{where } D(\underline{\pi_i}||\underline{\pi_j}) = \sum_{k=1}^{N} \pi_{ik} \log_2 \left(\frac{\pi_{ik}}{\pi_{jk}} \right). \tag{3.11}$$

The symmetric KL distance is separately obtained in each of the eight types of regions by training with synthetic data and testing with the camera images using the appropriately chosen coefficients as features. The overall divergence score is obtained by taking the mean of the individual divergence scores in eight regions and three color components. A low value of overall divergence score indicates that the two cameras are similar and are likely to use very similar kind of interpolation methods.

The divergence scores of the 19 different camera models are shown in Table 3.4. Here, the $(i, j)^{\text{th}}$ element in the matrix represents the average symmetric

KL distance between the interpolation coefficients of camera model$-i$ and camera model$-j$. Divergence scores below a threshold of 0.06 have been shaded. We observe from the table that most cameras from the same brand are likely to use similar kinds of interpolation algorithms. This is especially evident for some models of Canon and Minolta used in our analysis.

The divergence score between the two Canon models, S400 and S410, are very low, suggesting that both of these models are likely to use similar techniques for color interpolation. We also observe similarities between the two Minolta models, DiMage S301 and DiMage F100, and between the two Sony models, Cybershot DSC P7 and P72. The metric is close to zero in all these cases, thus indicating that cameras from the same manufacturer have similar interpolation. Interestingly, we also observe some similarity between several cameras from different manufactures. As shown in Table 3.4, the divergence score between Nikon model E4300 (camera no. 7) and the Minolta DiMage S304 (camera no. 13) is low, which suggests a resemblance in the type of interpolation used by these two cameras.

The work that we have presented so far quantifies the similarity of camera models based on the estimated color interpolation coefficients. The parameters of the other stages in the scene capture model, such as white balancing and JPEG compression, may be further used to study similarities among different camera models and brands. In such cases, the forensic information collected from various components may also be fused together to provide quantitative evidence to identify and analyze technology infringement/licensing of cameras.

3.4.4 Applications to Image Acquisition Forensics

The goal of image acquisition forensics is to determine the device type and the brand and model of the device that was used to acquire the image in question. In the previous sections, we have shown that the color interpolation coefficients can help identify the brand and model of the camera that was used to capture the image if indeed the image was originally camera captured. In this subsection, we extend the feature based classification approach to facilitate image acquisition forensics, and show that the proposed methods combined with noise features [54, 55] provide a very high accuracy in differentiating between images from different sources such as cell phones cameras, standalone cameras, scanners, and computer-graphics.

For our study, we use 100 images from each of the four scanner models (Epson Perfection 2450 photo, AcerScan, Canon CanoScan D1250U2F, and Microtek ScanMaker 3600), five different cell phone cameras models (Nokia 6102, Motorola V550, Samsung c417, Sony Ericsson W810, and Audiovox CDM-8910), and five standalone cameras models (Canon Powershot A75, FujiFilm Finepix S3000, Casio QV-UX2000, Minolta DiMage F100, and Canon PowerShot S410). A separate set of 100 computer graphics (CG) images were obtained from the Columbia university dataset [101]. The sample images were taken in completely random conditions, without any controlled experimental setup to simulate non-intrusive testing conditions. In this way, the image dataset simulates real-world data in terms of lighting, color, texture, and subject. The color interpolation coefficients and the noise features from [55] were estimated from each of the 1500 images in our database and employed for subsequent studies.

Table 3.5: Confusion matrix for device-type identification.

Device	Phone camera	Standalone Digital Camera	Scanner	Computer Graphics
Phone camera	93%	2%	0%	5%
Standalone camera	1%	98%	1%	0%
Scanner	1%	3%	94%	2%
Computer Graphics	4%	2%	4%	90%

Identifying Image Acquisition Device

For our study, 100 images from each device type (cell phone camera, standalone camera, and scanner) were selected with an equal number from each model, and all CG images were used, to create four classes of 100 images each. A randomly chosen set of 99 images from each class were used in training the SVM classifier, and the remaining image was used in testing to obtain the *leave-one-out* performance. The experiment was repeated 100 times with different set of training images and the average confusion matrix is shown in Table 3.5. Here, the $(i, j)^{\text{th}}$ element of the matrix corresponds to the fraction of images from source type$-i$ classified as belonging to source type$-j$. The main diagonal elements give the percentage of correct identification. From the results in Table 3.5, we find that overall identification accuracy is 93.75%, suggesting that the proposed features are good for identifying the source type.

Identifying Device Brand/Model

Once an image's source device has been determined, further analysis can be performed using the same set of features to identify the particular brand or model

of the device that was used to capture the image. In the previous subsections, we have presented results for camera brand and model identification and in this subsection, we focus on cell phone cameras and scanners. Finding the type of cell phone camera from its output images poses additional challenges, compared to standalone cameras and scanners, due to their lower image resolution, noisier image sensors, and a higher rate of default JPEG compression. In our results with cell phone cameras, we found that using interpolation coefficients alone, rather than a combination of interpolation coefficients and noise features, produced higher accuracies [94]. This result for cell phone cameras is expected because most cell phone camera brands/models employ different algorithms for color interpolation; and therefore, these coefficients alone provide tell-tale evidence to distinguish images from different brands/models. For our experiments with cell phone cameras, we used a randomly chosen 90 random images for training and the remaining 10 for testing, and the corresponding results are shown in Table 3.6. We find from the table that the average identification accuracy is close to 97.7% for five models, and this is significantly better than state-of-the-art techniques that produce average accuracies close to 92% over four camera models from two different camera brands [135].

We test the robustness of the proposed system for post-processing operations such as JPEG compression. To generate data, we compress the original cell phone camera images under different JPEG quality factors from 60% to 100%. The color interpolation coefficients are then obtained from the compressed images and used as features for classification. A randomly chosen 90 images were used in training the classifier and the remaining 10 were used in testing. The experiment was repeated 100 times and the average accuracies under different JPEG quality

Table 3.6: Confusion matrix for cell phone camera identification.

Cell Phone	Nokia	Motorola	Samsung	Sony	Audiovox
Nokia	95.8%	0.4%	0%	3.8%	0%
Motorola	2.8%	97.2%	0%	0%	0%
Samsung	1.2%	0%	97.8%	0.2%	0.8%
Sony	2.4%	0%	0%	97.6%	0%
Audiovox	0%	0%	0%	0%	100%

factor are shown in Figure 3.9. The figure shows that as the JPEG quality factor decreases, the identification accuracy decreases as expected. However, the lowest accuracy achieved is around 91% demonstrating the superior performance of the proposed features.

We compare the performance of the proposed features for cell phone camera identification with the higher order statistical features introduced in [36]. In our experiments with [36], we employ the same set of cell phone camera images (with 90 for training and 10 for testing) and examine the identification accuracies as a function of JPEG quality factors. The performance, averaged over 100 iterations, is shown alongside in Figure 3.9. The results suggest that the proposed features perform at least 12% better in identifying the cell phone brand/model, establishing the goodness of the proposed features.

For scanner identification, we found that using a combination of interpolation coefficients and noise feature parameters from [54] gave best results. 100 images from each of the four models of scanners were used, with 90 random images used for training and the remaining 10 used for testing. The overall identification accuracy for scanner brand was 96.2%. Further, the identification results were found to be

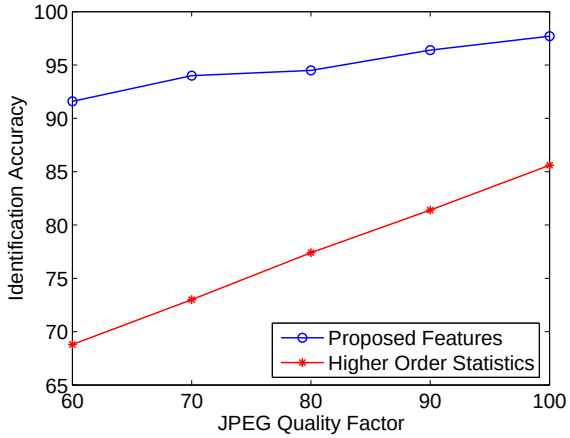


Figure 3.9: Robustness to JPEG compression for cell phone camera identification using (a) proposed color interpolation coefficients as features, and (b) higher order statistics [36] as features.

robust to moderate levels of post-processing operations such as JPEG compression, image sharpening, gamma correction, and contrast enhancement. Further details can be found in [54, 94].

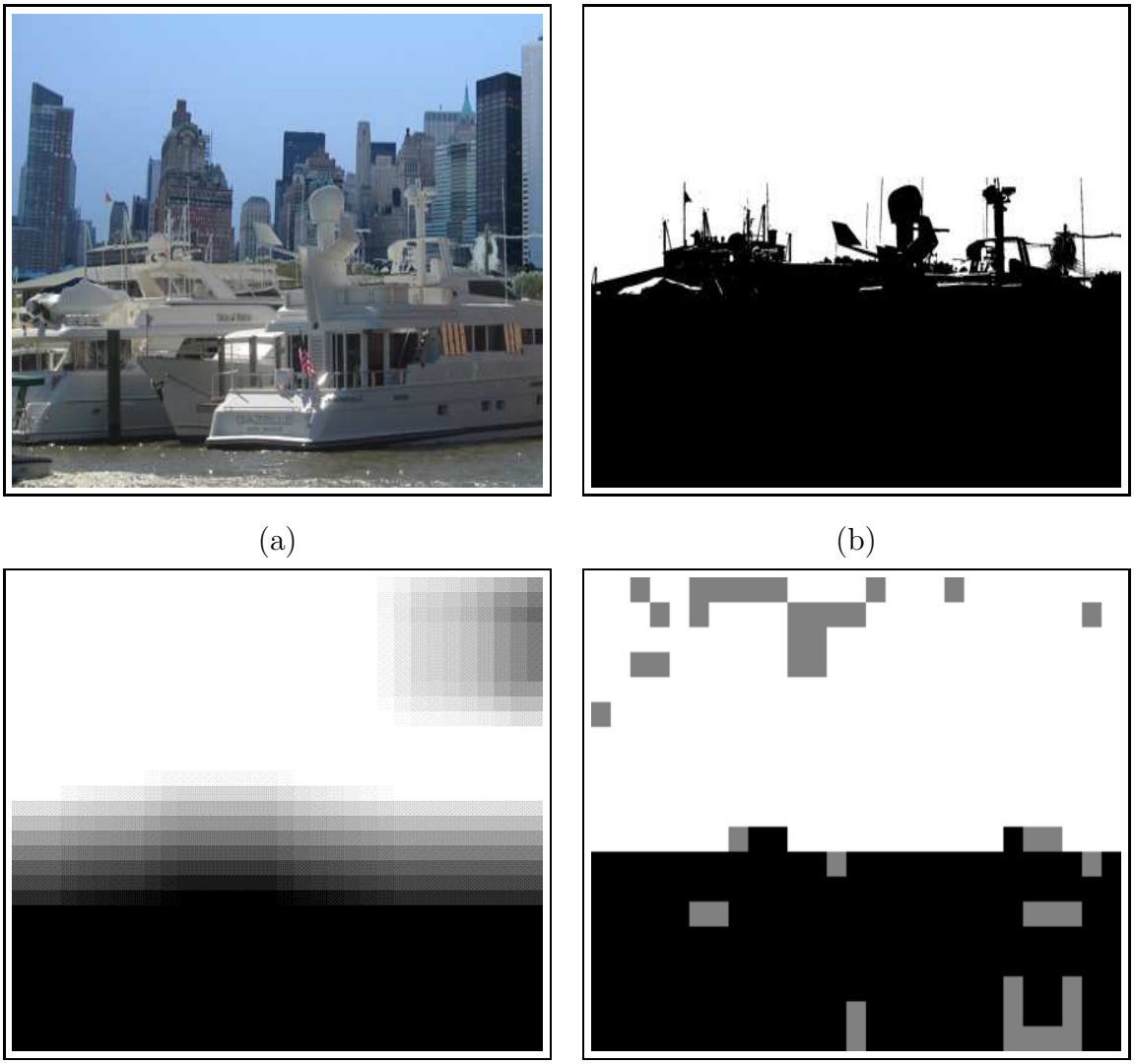
3.4.5 Detecting Cut-and-Paste Forgeries based on Inconsistencies in Component Parameters

Creating a tampered image by cut-and-paste forgery often involves obtaining different parts of the image from pictures captured using different cameras that may employ a different set of algorithms/parameters for its internal components. Inconsistencies in the estimated sensor pattern noise obtained from different regions of the image [86] or the inconsistencies in the estimated intrinsic fingerprint traces left behind by camera components [123] can be used to identify such digital forgeries as cut-and-paste operations. Here, we illustrate with a case study. We create

a tampered picture of size 2048×2036 by combining parts of two images taken using two different cameras. In Figure 3.10(a) and (b), we show the tampered picture and its individual parts marked with different colors. The regions displayed in white in Figure 3.10(b) are obtained from an image taken with the Canon Powershot S410 digital camera, and the black parts are cropped and pasted from a picture shot using the Sony Cybershot DSC P72 model. The combined image was then JPEG compressed with quality factor 80%.

To identify the intrinsic camera fingerprints in different parts of the picture, the image is examined using a sliding window of 256×256 with step size 64×64, and the color interpolation coefficients are estimated in each 256×256 block [123]. The $k-$means clustering algorithm [31] is then employed to cluster these features into two classes. With a step size of 64, each individual 64×64 sub-block would be analyzed 16 times to provide 16 different clustering results; the clustering results are represented as binary values (0 or 1) as labels for the two classes. Figure 3.10(c) shows the average of the clustering labels from these 16 sub-blocks. As shown in Figure 3.10(c), our results indicate that the features are clustered distinctly in two separate classes with the gray area in between representing the transition from one class to the other. In this particular case, we notice that the manipulated picture has tell-tale traces from two different cameras and is therefore tampered.

We then employ supervised training [31] using the 19-camera model classifier to further verify our results. The detection results from the 19-camera model classifier are shown in Fig. 3.10(d). In this figure, the regions marked black denotes those classified as the Sony Cybershot DSC P72 model and the white areas correspond to the parts correctly classified as the Canon Poweshot S410 model. The remaining regions represented in grey correspond to the blocks that were misclassified as one



(a) (b)

(c) (d)

Figure 3.10: Applications to source authentication showing (a) Sample tampered image; (b) Regions obtained from the two cameras; (c) Results from clustering the color interpolation coefficients with black representing Sony Cybershot DSC P72, white representing Canon Powershot S410 and shades of gray indicating the likelihood that the region is from Canon Powershot S410 with a value close to white denoting higher likelihood; (d) CFA interpolation identification results using the 19 camera-model classifier with black representing Sony Cybershot DSC P72, white representing Canon Powershot S410, and grey indicating the regions classified as other cameras.

of the remaining 17 camera models. As shown in Fig. 3.10(d), the results indicate that the correct camera can be identified with a very high confidence in most of the regions in the tampered picture using the data obtained from each 256×256 macro-block. In this particular case, we notice that the manipulated picture has distinct traces from two different cameras and is therefore tampered. A closer observation of the misclassified blocks (shown in grey) also indicates that most of these regions are clustered either around the tampering boundaries from two cameras or in very smooth areas of the image. Blocks around tampered regions would contain traces of both the camera models and thus might lead to incorrect classifications and misclassifications around the smooth regions of the image can be attributed to the fact that most cameras employ similar techniques such as bicubic interpolation around the smooth regions.

3.5 General Component Forensics Methodology

In this section, we extend the proposed non-intrusive forensic analysis to a methodology applicable to a broad range of devices. Let $O_1, O_2, \ldots, O_{N_o}$ be the sample outputs obtained from the test device that we model as a black box, and $C_1, C_2, \ldots, C_{N_c}$ be the individual components of the black box. Component forensics provides a set of methods to help identify the algorithm and parameters used by each of the processing blocks C_y. A general forensic analysis framework is composed of the following processing steps as shown in Figure 3.11.

1. *Modelling of the Test Device:* As the first step of forensic analysis, a model is constructed for the object under study. This modeling helps break down the test device into a set of individual processing components $C_1, C_2, \ldots, C_{N_c}$ and

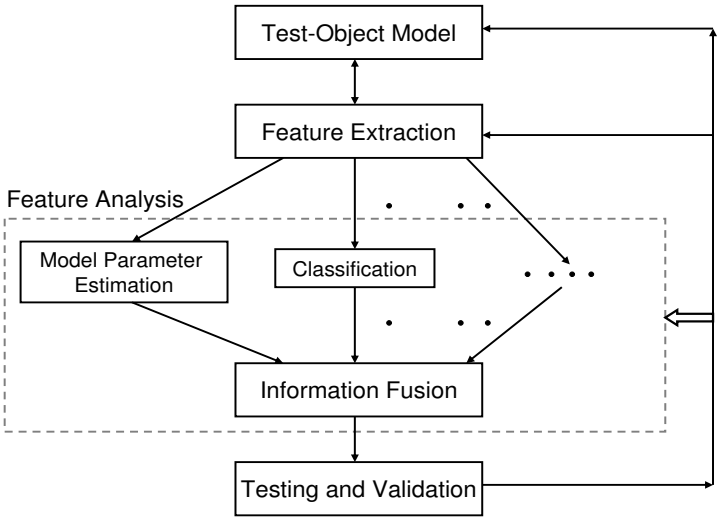


Figure 3.11: The proposed forensic analysis methodology.

systematically study the effect of each of these blocks on the final outputs obtained with the test object.

2. *Feature Extraction:* The forensic analyst identifies a set of features that has good potential to help identify the algorithms used in y^{th} device component C_y. These features are based on the final output data and are chosen to uniquely represent each of the algorithms used. For the case of digital cameras, we have used in this chapter the estimated color interpolation coefficients as features for forensic analysis. Parameters of other components, such as white balancing constants and gamma correction values, are also possible features to incorporate.

3. *Feature Analysis and Information Fusion:* We analyze the features extracted from the previous stage to obtain forensic evidence to meet specific applications' needs. The appropriate analysis technique depends on the component under study, the application scenario, and the type of evidence desired. The

results obtained from each of the analysis techniques can be combined to provide useful evidence about the inner working of the device components.

4. *Testing and Validation Process:* The validation stage uses test data with known ground truth to quantify the accuracy and performance of the forensic analysis system. It reflects the degree of success of each of the above processing stages and their combinations. Representative synthetic data obtained using the model of the test object can help provide ground truth to validate the forensic analysis systems and provide confidence levels on estimation. The results of this stage can also facilitate a further refinement of the other stages in the framework.

The methods and techniques adopted in each stage may vary depending on the device, the nature of the device components, and the application scenario. Regarding feature extraction, in some situations, the features by themselves (without further processing) can be proven to be useful forensic evidence and be used to estimate the parameters of the model. For instance, the color interpolation coefficients were directly estimated from the camera output, and used to study the type of interpolation in different regions of the image in Section 3.3.2. Evidence collected from such analysis can be used to study the similarities and differences in the techniques employed in the device components across several models and answer questions related to infringement/licensing and evolution of digital devices. In some other application scenarios, the component parameters might be an intermediate step and further processing would be required to answer specific forensic questions. For example, we have used the estimated color interpolation coefficients as features to build a robust camera identifier to determine the camera model (and make) that was used to capture a given digital image as seen in Sections 3.4.1 and

3.4.2.

3.6 Chapter Summary

In this chapter, we consider the problem of component forensics and propose a set of forensic signal processing techniques to identify the algorithms and parameters employed in individual processing modules in digital cameras. The proposed methodology is non-intrusive and uses only the sample data obtained from the digital camera to find the camera's color array pattern and the color interpolation methods. We show through detailed simulations that the proposed algorithms are robust to various kinds of postprocessing that may occur in the camera. These techniques are then used to gather forensic evidence on real world datasets captured with 19 camera models of nine different brands under diverse situations. The proposed forensic methodology is used to build a robust camera classifier to non-intrusively find the camera brand and model employed to capture a given image for problems involving image source authentication. Our results indicate that we can efficiently identify the correct camera brand with an overall average accuracy of 90% for nine brands. Our analysis also suggests that there is a considerable degree of similarity within the cameras of the same brand (e.g. Canon models) and some level of resemblance among cameras from different manufacturers. Measures for similarity are defined and elaborate case-studies are presented to elucidate the similarities and differences among several digital cameras. We believe that such forensic evidence would provide a great source of information for patent infringement cases, intellectual property rights management, and technology evolution studies for digital media.

Appendix I: Some Popular Color Interpolation Algorithms

There have been numerous algorithms employed in practice for Color Filter Array interpolation. In this appendix, we briefly review some of the popular methods. For a detailed survey, the readers are referred to [7]. Color interpolation methods can be broadly classified into two main categories, namely, adaptive and non-adaptive methods, depending on their adaptability to the image content. While non-adaptive methods use the same pattern for all pixels in an image, adaptive methods such as gradient based algorithms use the pixel values of the local neighborhood to find the best set of coefficients to minimize the overall interpolation error.

Bilinear and Bicubic methods are examples of non-adaptive interpolation schemes. In these algorithms, the pixel values are interpolated according to the following equation [112]:

$$S_{int}(x, y, c) = \sum_{u,v \in -N_g}^{N_g} h_c(u, v) S_{raw}(x - u, y - v, c),$$

where S_{raw} are the original raw values obtained from the sensor with $S_{raw}(., ., 1)$ representing the red color and so on, S_{int} denotes the interpolation results, and h_c denotes the 2-D filters of dimension $N_g \times N_g$ used in interpolation. In a general case, h_c may be dependent on the color channel. Let h_r, h_g, and h_b denote the values taken by h_c for red, green, and blue colors, respectively. For the bilinear case, these filters are given by

$$h_r = h_b = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \text{ and } h_g = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The corresponding filters for the bicubic case are given by

$$h_r = h_b = \frac{1}{256}\begin{bmatrix} 1 & 0 & -9 & -16 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ -16 & 0 & 144 & 256 & 144 & 0 & -16 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & -16 & -9 & 0 & 1 \end{bmatrix}, \quad h_g = \frac{1}{256}\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 1 & 0 & 81 & 256 & 81 & 0 & 1 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The *Smooth Hue* interpolation algorithm is based on the observation that the hue varies smoothly in natural images. In this algorithm, the green channel is first interpolated using bilinear interpolation to yield $S_{int}(.,.,2)$. The red components are then obtained by interpolating the ratios of 'red/green' via

$$\frac{S_{int}(x,y,1)}{S_{int}(x,y,2)} = \frac{1}{2}\left(\frac{S_{raw}(x,y-1,1)}{S_{int}(x,y-1,2)} + \frac{S_{raw}(x,y+1,1)}{S_{int}(x,y+1,2)}\right).$$

The blue components can be obtained similarly by interpolating the 'blue/green' ratios.

In *Median* filter based algorithms, the three channels are first interpolated using bilinear interpolation. Then the differences 'red−green', 'red−blue', and 'green−blue', are median filtered to produce M_{rg}, M_{rb}, and M_{gb}, respectively. At each pixel location, the missing color values are obtained by linearly combining the original color sensor value and the appropriate median filter result [112]. For example, the green color component at the location of the red color filter is obtained as

$$S_{int}(x,y,2) = S_{raw}(x,y,1) - M_{rg}(x,y).$$

All the methods described above are non-adaptive in nature and do not depend on the characteristics of particular regions. In contrast to these techniques, the *Gradient Based* algorithms [76] are more complex. Here, the horizontal gradient

(J_h) and the vertical gradient (J_v) at the point (x, y) are first estimated using

$$\begin{aligned} J_h(x,y) &= |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|, \\ J_v(x,y) &= |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|, \end{aligned}$$

where $I_{x,y} = S_{raw}(x, y, p(x, y))$ and p is the CFA pattern matrix (e.g. Bayer pattern) with $p(x, y) = 1, 2$, or 3, indicating that the CFA pattern at the $(x, y)^{\text{th}}$ pixel is red, green, or blue, respectively. The edge direction is then estimated from the gradient values, and the missing pixel values in the green component of the image are obtained in such a way that the interpolation is done along the edge and not across the edge, using only pixel values from the green channel. The missing red and blue components are found by interpolating the difference, 'red−green' and 'blue−green' along the edge, respectively.

The *Adaptive Color Plane* interpolation method [56] is an extension of the gradient based method. Here, the horizontal and vertical gradients are estimated using

$$\begin{aligned} J_h(x,y) &= |I_{x,y-1} - I_{x,y+1}| + |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|, \\ J_v(x,y) &= |I_{x-1,y} - I_{x+1,y}| + |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|. \end{aligned}$$

Unlike the simple gradient based method, the interpolation of one color component here also uses the other colors, and the output is a linear combination of sampled sensor outputs in the neighborhood across the three color channels [56].

Appendix II: Probabilistic Support Vector Machines

We employ the probabilistic SVM framework proposed in [148] to find the likelihood q_i that a given data sample comes from the i^{th} class. Let the observation feature vector be denoted as $\mathbf{x}$ and the class label as y, where $1 \leq y \leq c$ for a c-class problem. With the assumption that the class-conditional densities $Pr(\mathbf{x}|y)$ are exponentially distributed [108], the estimate $\hat{\mu}_{ij}$ of the pairwise class probabilities $\mu_{ij} \triangleq Pr(y = i|y = i \text{ or } j, \mathbf{x})$ is found by fitting a parametric model to the posterior probability density functions $\hat{\mu}_{ij} = 1/(1 + \exp(\hat{\mathbf{a}}\mathbf{x} + \hat{b}))$. The values of $\hat{\mathbf{a}}$ and $\hat{b}$ are estimated by minimizing the *Kullback-Leibler* distance between the parametric pdf define earlier and the one observed obtained from the training samples. We then find $q_i \triangleq Pr(y = i|\mathbf{x})$, the probability that the data sample comes from the i^{th} class for a $c-$class SVM, by solving the optimization problem that minimizes the following:

$$\min_{q_1,q_2,\ldots,q_c} \sum_{i=1}^{c} \left(\sum_{j,j\neq i} (1 - \hat{\mu}_{ij}) q_i - \sum_{j,j\neq i} \hat{\mu}_{ij} q_j \right)^2$$

$$\text{subject to} \quad \sum_{i=1}^{c} q_i = 1, \;\; q_i \geq 0, i = 1, 2, \ldots, c.$$

Further details of the algorithm can be found in [148], and a possible implementation is available at [22].

Chapter 4

Digital Image Forensics via Intrinsic Fingerprints

In Chapter 3, we showed that any change or inconsistencies in the estimated intrinsic fingerprints can help detect forgeries. In this chapter, we take a closer look at approaches for tampering detection and steganalysis and introduce a new methodology for digital image forensics of color images aimed at identifying different types of global tampering operations. The algorithm works in a two steps. In the first step, using a detailed imaging model and its component analysis as presented in Chapter 3, we estimate the intrinsic fingerprints of the various *in-camera* processing operations. We then model any further processing applied to camera outputs as a filtering operation, and estimate its coefficients to obtain the *post-camera* fingerprints. We show that absence of estimated in-camera fingerprints suggests that the test image is not a camera output and is possibly generated by other image production processes, and any change or inconsistencies among the estimated in-camera fingerprints, or the presence of new post-camera fingerprints indicates that the image has undergone some kind of post-camera processing. We begin this

chapter by reviewing related work in Section 4.1. The proposed forensic framework to estimate the post-camera fingerprints is presented in Section 4.2. Detailed simulation results and elaborate case studies are then presented in Section 4.3 and Section 4.4, respectively, and the chapter is summarized in Section 4.5.

4.1 Related Work on Tampering Detection and Steganalysis

In the forgery detection literature, there have been work that try to address the problem of identifying if the given digital image has gone through any processing, such as tampering or steganographic embedding, after being produced by the camera. These work try to define the properties of a manipulated image in terms of the distortions it goes through, and using such analysis present methods for detecting manipulated images. For instance, some work assume that creating a tampered image involves a series of processing operations, which might include resampling [111], JPEG compression [33,80,84], Gamma correction [34], and chromatic aberration [67]. Based on this observation, they propose to identify such manipulations by extracting certain salient features that would help distinguish such tampering from authentic data.

When the image is upsampled, some of the pixel values are directly obtained from the smaller version of the image, and the remaining pixels are interpolated and thus highly correlated with its neighbors. Thus, post-processing operations such as resampling can be identified by studying the induced correlations [111]. JPEG compression has been considered as quantization in the discrete cosine transform (DCT) domain and statistical analysis based on binning techniques have been

used to estimate the quantization matrices [33, 84]. Image manipulations such as contrast changes, Gamma correction and other image non-linearities have been modelled and higher order statistics such as the bispectrum have been used to identify and blindly correct them [37, 110]. Inconsistencies in noise patterns [110], JPEG compression [35], or lighting [66], and alternations in correlations induced by color interpolation [112] caused while creating a tampered picture have been used to identify inauthentic images.

In [35], the authors exploit the differences in JPEG quantization tables among different cameras to introduce an image authentication scheme. Given a digital image, the authors first estimate the quantization table from it [84, 110], and then compare the estimated tables with a database of quantization tables collect apriori from 204 different digital cameras. A mis-match in the estimated quantization tables with the ones in the database or across different regions of the image suggests that the image has been manipulated after being captured by a digital camera. Johnson *et al.* model inconsistencies in lighting directions to determine possible tampering [66]. A 2-D model is constructed based on an earlier work by Nillius *et al.* [103] and the lighting direction is estimated non-intrusively from the image based on this model. The authors show through simulations that the lighting direction can be estimated up to an error of two degrees when tested with infinite, local, and multiple light sources; therefore, assisting in the detection of contradicting light sources.

Although these methods can be employed to identify the type, and the parameters of the post-processing operation, it would require an exhaustive search over all the numerous kinds of post-processing operations to detect tampering. Based on this observation, blind tampering detection methods based on sensor

noise patterns and image features were proposed. The presence of pattern noise in camera-captured images and its absence in tampered images have been used to detect forgeries [86]. Ng *et al.* employed bicoherence features to detect the presence of abrupt discontinuities in the image and use such analysis to detect tampering and to distinguish between photographic and computer graphics images [102]. In [36], the authors show that wavelet features extracted from the image can be employed for other forensic applications including distinguishing between natural and un-natural synthetic images, plan text and stego data, computer graphics images and photographs, and differentiating between live and broadcast images [36, 87]. Avcibas *et al.* develop a set of content independent features based on analysis of variance approaches and image quality metrics [10] for distinguishing between unmanipulated images and images manipulated via brightness or contrast enhancement [8]. These methods [8, 36, 53] require samples of tampered images for classification to distinguish manipulated images from genuine ones. Further, these methods may not be able to efficiently identify other kinds of manipulations that are not modelled or considered directly. By defining the properties of an authentic image via intrinsic fingerprints, our proposed methods provide better scalability and can help identify previously unseen distortions as will be seen in Section 4.3.

In steganalysis literature, there have been work that identify the presence of hidden information in multimedia data. These work can be broadly classified into two classes, namely embedding-specific and universal. In the class of embedding-specific steganalysis, there have been algorithms to identify different types of least significant bit (LSB) embedding [43, 45, 143]. Statistics based approaches for universal blind staganalysis have been introduced in [9, 88], where features from wavelet statistics [88] or image quality measures [9] are used to build

a classifier to distinguish stego data from cover data.

Most of these techniques mentioned above are primarily targeted at finding the processing steps that occur after the image has been captured by the camera, and are not for finding the algorithms and parameters used in various components inside the digital camera. As shall be seen from our results, the proposed forensic methodology based on intrinsic fingerprints provides a combined framework for authenticating digital camera outputs and distinguishing them from scanned, computer generated, tampered, and stego data.

4.2 Estimating Intrinsic Fingerprints of Post-Camera Manipulations

In this section, we present methods to estimate the intrinsic fingerprints of post-camera manipulations under the assumption that the entire image has undergone the same manipulation. This approach can be extended over a block-by-block basis to estimate the intrinsic fingerprints in individual blocks. Given a test image or an image block, S_t, we introduce a non-intrusive forensic methodology to identify if it has undergone any further processing after it is being captured using a digital camera. We first assume that S_t is a manipulated camera output corresponding to the *point B* in Figure 2.3, and is obtained by processing the actual camera output S_d (*point A* in Figure 2.3) using the manipulation block. We then represent the post-camera processing applied on S_d as a combination of linear and non-linear operations, and approximate them with a linear shift-invariant filter. The coefficients of this *manipulation filter*, estimated using blind deconvolution, serve as our post-camera fingerprints to answer a number of forensic questions related to

the origin and the authenticity of digital images [132]. In the following subsections, we describe the estimation algorithm in detail.

4.2.1 Computing Inverse Manipulation Filter Coefficients by Constrained Optimization

Let S_t denote the test image, and let S_{te} represent the estimate of the camera output obtained by passing the given test image through the inverse manipulation filter u, *i.e.*,

$$S_{te}(x, y, c) = \sum_{m,n} u(m, n, c) S_t(x - m, y - n, c), \text{ for } 1 \leq c \leq 3. \tag{4.1}$$

Here, we assume that $u(.,.,.)$ is of dimension $N_u \times N_u \times 3$, and operates independently on each color component. The coefficients of the inverse manipulation filter, u, are estimated by solving an optimization problem that minimizes the camera model fitting error, $E(u)$, given by

$$E(u) = \sum_{c=1}^{3} \sum_{x,y} \left(\hat{S}_{te}(x, y, c) - \sum_{m,n} u(m, n, c) S_t(x - m, y - n, c) \right)^2, \tag{4.2}$$

where $\hat{S}_{te}$ denotes the image formed from S_{te} by imposing the constraints that pixels from a camera output image should satisfy due to CFA based color interpolation:

$$\hat{S}_{te}(x, y, c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m, n, c) S_{te}(x - m, y - n, c) \\ \qquad\qquad \forall \{x, y\} \in \Re_i, \text{ and } 1 \leq c \leq 3, \\ S_{te}(x, y, c) \qquad\qquad \text{otherwise.} \end{cases} \tag{4.3}$$

In these *camera constraints*, $\alpha_{\Re_i}$ denote the estimates of the color interpolation coefficients, and are derived from the image S_{te} using the component forensics techniques presented in Section 3.2 . In our work, we assume that $\sum_{m,n} u(m, n, c) = 1$

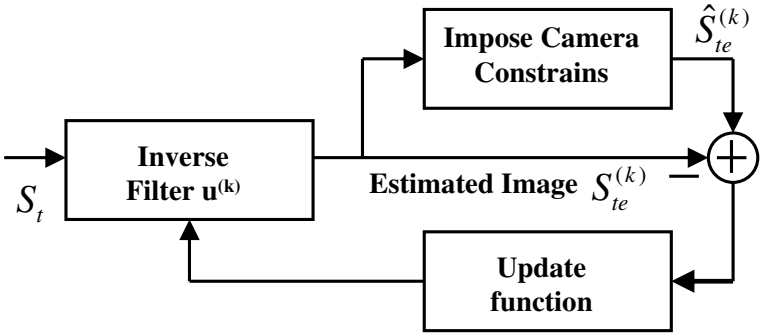


Figure 4.1: Recursive algorithm to estimate the coefficients of the manipulation filter.

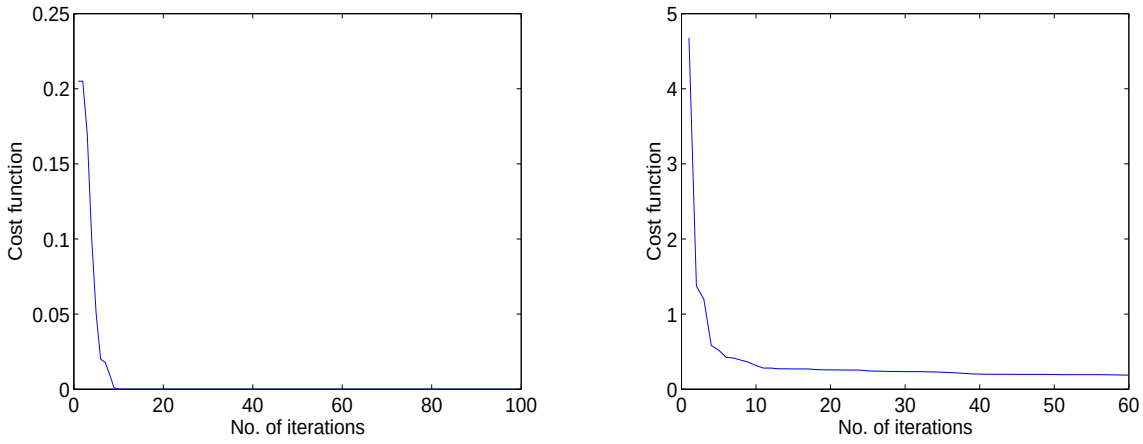
for $c = 1, 2, 3$ to ensure that the original image and its manipulated version have similar brightness levels. Incorporating this gain constraint into the minimization problem, we solve for u by minimizing a modified cost function, $J(u)$, given by

$$J(u) = \sum_{x,y,c} \left(\hat{S}_{te}(x, y, c) - \sum_{m,n} u(m, n, c) S_t(x - m, y - n, c) \right)^2 + \eta \sum_{c=1}^{3} \left(\sum_{m,n} u(m, n, c) - 1 \right)^2, \quad (4.4)$$

where the value of η is chosen to adjust the weights of the relative individual costs.

The filter coefficients can be directly estimated in the pixel domain through a recursive procedure illustrated in Figure 4.1. We start the iteration by setting $u^{(0)}$ to be a delta function; this corresponds to direct camera outputs. In the k^{th} iteration, we obtain an estimate of the camera output, $S_{te}^{(k)}$, by passing the test image S_t though the estimate of the inverse blur filter $u^{(k)}(.,.,.)$. We then impose camera constraints given by (4.3) to get $\hat{S}_{te}^{(k)}$ and find the camera model fitting error. The inverse filter coefficients are then updated [74] by

$$u^{(k+1)} = u^{(k)} + t_k d_k, \quad (4.5)$$



(a) (b)

Figure 4.2: Convergence of the cost function for (a) unmanipulated image, (b) manipulated image filtered with a 5×5 averaging filter.

where

$$d_k = \begin{cases} -\nabla J(u^{(k)}), & \text{if k} = 0, \\ -\nabla J(u^{(k)}) + \lambda_{k-1} d_{k-1}, & \text{otherwise}, \end{cases} \tag{4.6}$$

$$\lambda_{k-1} = \frac{< \nabla J(u^{(k)}) - \nabla J(u^{(k-1)}), \nabla J(u^{(k)}) >}{||\nabla J(u^{(k-1)})||^2}, \tag{4.7}$$

and the step sizes t_k are chosen as the one that minimizes $J(u^{(k)} + t_k d_k) \leq J(u^{(k)} + t d_k)$ for all t. The recursive procedure is repeated for a finite number of iterations or until convergence. In the Appendix of this chapter, we show that the optimization problem is convex and converges to a unique solution for all images whose interpolation parameters $\alpha_{\Re_i}$ can be estimated accurately.

We test the blind deconvolution method for a sample direct camera output along with its filtered versions. Figure 4.2(a) and (b) shows the variation of the modified cost function J given by (4.4) as a function of the number of iterations for a sample unmanipulated image and an image filtered with an 5×5 averaging

–0.077	0	–0.077	0	–0.077
0	–0.077	0	–0.077	0
–0.077	0	1.923	0	–0.077
0	–0.077	0	–0.077	0
–0.077	0	–0.077	0	–0.077

(a)

0.129	0.041	–0.051	0.058	0.134
0.056	0.156	–0.273	0.159	0.022
–0.044	–0.281	0.764	–0.274	–0.032
0.026	0.156	–0.276	0.155	0.059
0.139	0.061	–0.049	0.034	0.125

(b)

Figure 4.3: Estimated inverse manipulation filter coefficients for (a) unmanipulated image, (b) manipulated image filtered with a 5×5 averaging filter. The inverse filter kernel size is set to 5×5.

filter, respectively. We observe that the cost function converges in 10 iterations in both cases. The final estimated inverse filter coefficients $u(.,.,2)$ for the green color channel for the two cases are shown in Figure 4.3(a) and (b), respectively. While the estimated coefficients from the unmanipulated camera output in Figure 4.3(a) are very close to an identity transform (corresponding to no post-camera manipulations), the corresponding manipulation coefficients derived from the average filtered image, as presented in Figure 4.3(b), are similar to the 5×5 kernel approximation of the inverse of the 5×5 averaging filter.

The performance of the blind deconvolution algorithm for tampering detection is to a great extent tied with the choice of the kernel size. In an ideal scenario, a finite size averaging filter in the pixel domain would require an infinite length kernel for its inverse. Although a larger kernel gives enhanced performance improvements, it requires more iterations for convergence. In the next subsection, we present a solution to directly estimate the filter coefficients in frequency domain.

4.2.2 Estimating Manipulation Filter Coefficients by Iterative Constraint Enforcement

The recursive algorithm described in Figure 4.1 can be solved in the frequency domain to directly obtain the manipulation filter coefficients by iteratively applying known constraints to the input image [11]. A schematic diagram of the *iterative constraint enforcement* algorithm is shown in Figure 4.4. The test image S_t is used to initialize the iterative process. In each iteration, the estimated camera output, g, and the estimated filter coefficients, h, are updated by repeatedly applying known constraints on the image and the filter in the pixel domain and the Fourier domain. In the k^{th} iteration, the *pixel domain constraints* on the image g_k consists of

1. *Real-valued constraints* that enforce the image pixel values to be real,
2. *Boundedness constraints* restricting the image pixel values to the range $[0, 255]$, and
3. *Camera constraints* of CFA-based color interpolation given by

$$\hat{g}_k(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) g_k(x-m, y-n, c), \\ \qquad\qquad \forall \{x,y\} \in \Re_i, \text{ and } 1 \leq c \leq 3 \\ g_k(x,y,c) \qquad\qquad \text{otherwise,} \end{cases} \tag{4.8}$$

where $\alpha_{\Re_i}$ denote the estimates of the color interpolation coefficients derived from the image g_k using the component forensics techniques presented in Section 3.2. After the image $\hat{g}_k$ is obtained, it is transformed by Discrete Fourier transform (DFT) to give $\hat{G}_k$. The frequency response H_k of the estimated manipulation filter in the k^{th} iteration is obtained using the technique described in [72, 73] with

$$H_k = \frac{\mathcal{F}(S_t)\hat{G}_k^*}{|\hat{G}_k|^2 + \frac{\beta_1}{|H_{k-1}|^2}}, \tag{4.9}$$

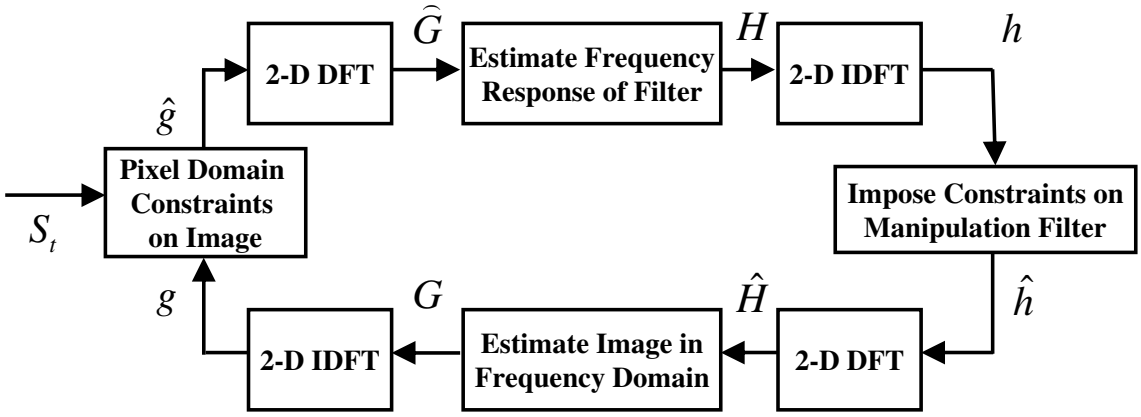


Figure 4.4: Schematic diagram of the iterative constraint enforcement algorithm.

where β_1 is an appropriately chosen constant, $\mathcal{F}(S_t)$ denotes the Fourier transform of the test image S_t, and $\hat{G}_k^*$ represents the complex conjugate of $\hat{G}_k$. The value of H_0 for the first iteration is initialized as $H_0 = \mathcal{F}(S_t)/\hat{G}_0$. The estimated filter response H_k is then inverse Fourier transformed to give h_k. We further impose *filter constraints* on h_k and obtain $\hat{h}_k$ to be the real part of h_k. The value of G_{k+1} for the $(k+1)^{\text{st}}$ iteration is obtained as a function of its two available estimates, (a) previous value, G_k, and (b) the estimate obtained by enforcing the Fourier domain constraint, $(FS_t/\hat{H}_k)$, where $FS_t = \mathcal{F}(S_t)$ and $\hat{H}_k = \mathcal{F}(\hat{h}_k)$. Both these estimates have their unique properties – G_k has a non-negative inverse transform that satisfies the image domain constrains, and $(FS_t/\hat{H}_k)$ satisfies the Fourier domain constraints. In our work, we average these two estimates separately in every iteration for each spatial frequency value and color to obtain the new estimate for G_{k+1} as described in [11]:

$$G_{k+1} = \begin{cases} G_k & \text{if } |FS_t| < \gamma, \\ (1-\beta_2)G_k + \beta_2 \frac{FS_t}{\hat{H}_k} & \text{if } |FS_t| \leq |\hat{H}_k| \text{ and } |FS_t| \geq \gamma, \\ \left(\frac{(1-\beta_2)}{G_k} + \frac{\beta_2 \hat{H}_k}{FS_t}\right)^{-1} & \text{if } |FS_t| > |\hat{H}_k| \text{ and } |FS_t| \geq \gamma. \end{cases} \tag{4.10}$$

Here, γ and β_2 are appropriately chosen constants. The value of γ represents the noise resilience of the system, and β_2 is chosen to lie in the range $[0, 1]$ to indicate the relative significance of the two terms in update equation [11]. In our experiments, we set $\gamma = 10^{-5}$ and $\beta_1 = \beta_2 = 0.3$. Finally, G_{k+1} is inverse Fourier transformed to give g_{k+1}, the pixel domain estimate of the camera output image, and the system proceeds to the next iteration. This process is repeated for a finite number of iterations and the frequency response of the estimated manipulation filter parameters H are found, to obtain the intrinsic fingerprints of post-camera manipulations. Deviation of the estimated manipulation filter parameters from an identity transform indicates that the test image has been manipulated after capture by the camera.

4.2.3 Performance Studies on Detecting Manipulations with Synthetic Data

We use synthetic data constructed from 100 representative images to study the performance of the blind deconvolution techniques for tampering detection [124, 132]. These 100 images are first down-scaled by a factor of 2×2 to remove the effects of previously applied filtering and interpolation operations, sampled on the Bayer filter [6, 7] array and then interpolated using six different interpolation algorithms to reproduce the scene capture process in cameras. For our simulations, we consider six different color interpolation methods: (a) bilinear, (b) bicubic, (c) smooth hue, (d) median filter, (e) gradient based, and (f) adaptive color plane. Details about these interpolation algorithms can be found in [7]. These 600 images that satisfy the camera model form our unmanipulated set. Processed versions are then obtained by applying average filtering to these 600 images with different filter

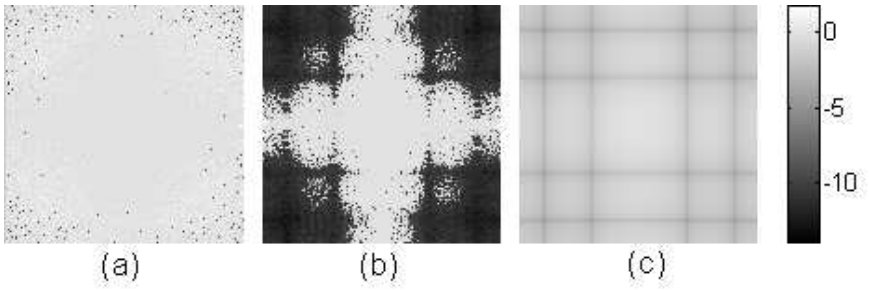


Figure 4.5: Frequency response of the manipulation filter for (a) A simulated unmanipulated camera output, and (b) Image low-pass filtered with a 5×5 averaging filter; (c) Actual manipulation filter coefficients of the 5×5 averaging filter shown alongside for comparison. The magnitude of the frequency response is shown in the $\log_{10}$ scale.

orders from 3 to 11.

We run the proposed blind deconvolution methods on all the images and compute the coefficients of the manipulation filter in each case using iterative constraint enforcement algorithm. In Figure 4.5(a), we show the estimated Fourier transform for a simulated unmanipulated camera output. We notice that it is almost a constant flat spectrum, representing an identity transform. The corresponding estimated frequency response for a 5×5 average filtered image is shown in Figure 4.5(b), and the actual coefficients are shown in Figure 4.5(c) for comparison. The similarity among the estimated and the actual coefficients justifies the performance of the the blind deconvolution algorithms.

A closer look at the frequency response of the manipulation filter for an unmanipulated camera output, shown in Figure 4.5(a), suggests minor deviations from an ideal flat spectrum. These deviations are attributed to the various post-interpolation processing that are done inside the cameras such as compression, denoising, and white balancing. To compensate for these minor deviations, we

use the spectral response H_{ref}, obtained using the blind deconvolution algorithm, from an authentic camera output as reference. Given the test input S_t, we find the frequency domain coefficients of the manipulation filter H_t and compare it with H_{ref} to measure the similarity among the coefficients. More specifically, we first find $\Theta_t = \log_{10}(|H_t|)$ to obtain the logarithm of the magnitude of the frequency response, and compute the similarity between the coefficients of the test input and the reference image using the similarity score defined as

$$s(\Theta_t, \Theta_{ref}) = \sum_{m,n} (\Theta_t(m,n) - \mu_t) \times (\Theta_{ref}(m,n) - \mu_{ref}), \tag{4.11}$$

where μ_t denotes the mean of the Θ_t, and μ_{ref} represents the mean of the Θ_{ref}. The test input is then classified as unmanipulated if the similarity to the reference pattern is greater than a suitably chosen threshold. On the other hand, if the input image has undergone tampering or steganographic embedding operations, the estimated manipulation filter coefficients would include the effects of both the post-camera manipulation operations along with post-interpolation processing inside the camera. In this case, the manipulation filter coefficients would be less similar to the reference pattern, and the similarity score would be lower than the chosen threshold.

We examine the performance of the *threshold based classifier* in terms of the receiver operating characteristics (ROC) [124]. For each original image, we compute the frequency response of the equivalent manipulation filter and measure its similarity with the reference filter pattern. The fraction of original images with a similarity score lower than a threshold τ is found to give the false alarm probability P_F. Similarly, we record the fraction of manipulated images (filtered in this case) with a similarity score less than τ to give the probability of correct decision P_D. We repeat this process for different decision thresholds τ, and arrive at the ROC as

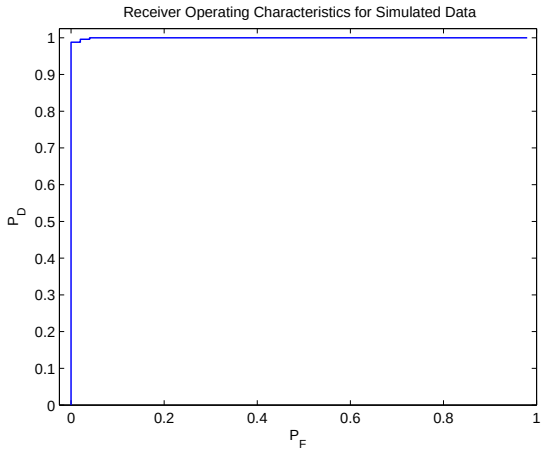


Figure 4.6: Receiver operating characteristics for distinguishing between simulated camera outputs and its filtered versions.

shown in Figure 4.6. We observe from the figure that the proposed scheme attains a $P_D \approx 1$ for $P_F = 0$. This suggests that the proposed scheme can effectively distinguish between direct camera outputs and its filtered versions.

4.3 Detecting Tampering on Camera Captured Images

Forensic evidence obtained by analyzing the coefficients of the manipulation filter provides clues about possible image tampering. Most often, creating a realistic tampered image involves a series of post-camera processing operations such as filtering, compression, resampling, contrast change, and others, that may be applied globally to the entire image or locally to different regions of the image. These processing operations leave distinct traces in the final picture and can be detected using the threshold based classifier by comparing the estimated manipulation filter coefficients with the reference pattern. In this section, we study the performance of

the proposed techniques for detecting different types of global image manipulations with real camera data. The forensic methodologies discussed in this section can be extended to detect local tampering by applying the techniques on a block-by-block basis.

4.3.1 Simulation Setup

A total of nine camera models as shown in Table 4.1 are used in our experiments. For each of the nine camera models, we have collected about 100 images. The images from different camera models are captured under uncontrolled conditions−different sceneries, different lighting situations, and compressed under different JPEG quality factors as specified by default values in each camera. The default camera settings (including image size, color correction, auto white balancing, and JPEG compression) are used in image acquisition. From each of these images, we randomly crop a 512×512 portion and use it for subsequent analysis. Thus, our *camera image database* consists of a total of 900 different 512×512 pictures. These images were then processed to generate 21 tampered versions per image to obtain 18900 manipulated images, and the 21 manipulation settings are listed in Table 4.2.

4.3.2 Classification Methodology and Simulation Results

We study the discriminative capabilities of our proposed schemes in terms of the ROC of the hypothesis testing problem with the following two hypotheses:

- Υ_0: image is a direct camera output,
- Υ_1: image is not a direct camera output and is possibly manipulated.

Table 4.1: Camera models used in experiments.

No.	Camera Model	No.	Camera Model
1	Canon Powershot A75	6	Canon EOS Digital Rebel
2	Canon Powershot S410	7	Nikon E4300
3	Canon Powershot G6	8	Fujifilm Finepix S3000
4	Canon Powershot S400	9	Sony Cybershot DSC P72
5	Canon Powershot S1 IS		

For each image, we compute the frequency domain coefficients of the estimated manipulation filter and determine its similarity with the chosen reference pattern. Images with a similarity score greater than a threshold are classified as authentic.

To choose the reference pattern, we randomly select a set of N_t training images along with its manipulated versions in the training stage. Using each of these N_t images, we compute the *in-class* and *out-class* similarity scores. More specifically, given the i^{th} image ($1 \leq i \leq N_t$), we calculate the *in-class* similarity scores by comparing the manipulation filter estimated from the i^{th} image and the estimates obtained from the remaining $(N_t - 1)$ images using (4.11). The *out-class* scores are then found by quantifying the similarity among the manipulation filter of the i^{th} image and the filter coefficients derived from the remaining tampered images. Using a threshold τ, the fraction of direct camera outputs with a similarity score lower than τ is computed to give the false alarm probability $P_F = \Pr(\Upsilon_1|\Upsilon_0)$, and the fraction of manipulated images with a similarity score less than τ is found to give the probability of correct decision $P_D = \Pr(\Upsilon_1|\Upsilon_1)$. We repeat this process for different decision thresholds τ to arrive at the ROC, and compute the area under the curve. These steps are performed separately with each of the N_t images in

Table 4.2: Tampering operations included in the experiments.

Manipulation Operation	Parameters of the Operation	Number of Images
Spatial Averaging	Filter orders 3-11 in steps of 2	5
Median Filtering	Filter orders {3, 5, 7}	3
Rotation	Degrees {5, 10, 15, 20}	4
Resampling	Scale factors {0.5, 0.7, 0.85, 1.15, 1.3, 1.5}	6
Additive Noise	PSNR 5dB and 10 dB	2
Histogram Equalization		1
Total		21

the training stage, and the manipulation filter coefficients that gives the maximum area under the ROC curve is chosen as the *reference pattern*. After choosing the reference pattern in the training stage, we compute the in-class and out-class similarity scores by comparing the chosen reference pattern with the filter coefficients obtained from the remaining camera outputs and its corresponding tampered versions, respectively, in our database in the testing stage. The corresponding ROC curves are obtained through this process.

Testing with Images from Canon Powershot A75

We test the performance of the proposed techniques using the 100 images from Canon Powershot A75. We choose this camera for two reasons: (a) based on our experimental studies, we observe that a linear shift-invariant model for the color interpolation coefficients fits well with the cameras' interpolation in each type of region and gives a very low fitting error; and (b) we observe that this Canon camera

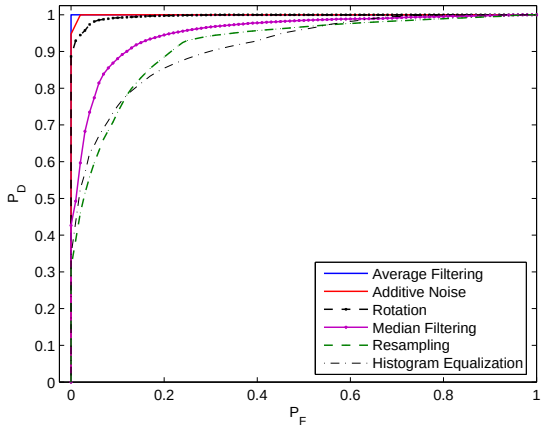


Figure 4.7: Receiver operating characteristics for tampering detection for images from Canon Powershot A75 when 50 images are used in training and the remaining 50 images are used in testing.

uses the same JPEG quantization table for all images that it captures, invariant of the input scene. Therefore, all images from the camera undergo the same kind of post-processing operations after color interpolation.

For our analysis with images from Canon Powershot A75, we use a randomly chosen set of 50 images for training, and test on the remaining 50 images along with the corresponding 50×21 tampered images. Figure 4.7 shows the performance of the threshold based detector averaged over 100 iterations. At relatively low P_F around 10%, the probability of correct detection is about $80\% - 95\%$ for most types of manipulations tested. Here, the results are based on a two-class classification problem, wherein the first class includes the direct camera outputs and the second class consists of camera outputs that have undergone a specific type of manipulation.

Testing with Diverse Inputs from Multiple Cameras

We now examine the performance of the proposed techniques under diverse input conditions. More specifically, we use all the 900 direct camera output images for the untampered dataset. These images were captured under the default camera settings and may have undergone different kinds of in-camera post-processing operations such as JPEG compression after color interpolation.

Figure 4.8 shows the ROC curve for detecting each manipulation. Here, we use a randomly chosen set of 200 images to train the classifier and test with the remaining 700 images; the experiments are repeated over 100 times to obtain an average ROC curve. In this case, we observe that for P_F close to 10%, the probability of correct detection is close to 100% for such manipulations as spatial averaging and additive noise, and around $70\% - 80\%$ for median filtering, histogram equalization, and rotation. These results are better than other work in the literature that are applicable to blind tampering detection [36, 112].

Comparing the results in Figure 4.8 with the results with the Canon Powershot A75 in Figure 4.7, we notice around $5\% - 10\%$ performance drop in detection accuracy for the same false positive rate. This reduction in performance can be attributed to the different types of post-processing operations performed after color interpolation in various camera brands and models. In our future work, we plan to estimate the parameters of such post-interpolation operations as JPEG compression [84] and white balancing, and include them into the system model to bridge the performance gap.

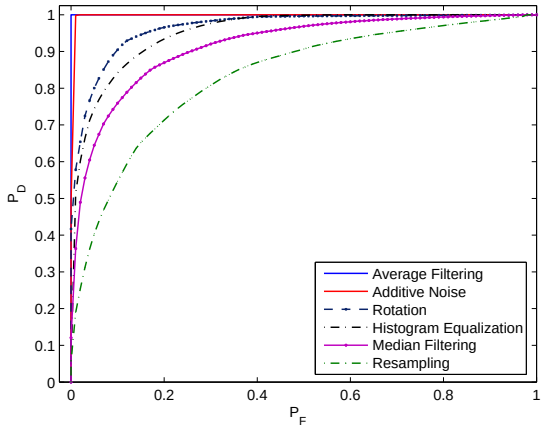


Figure 4.8: Receiver operating characteristics for tampering detection when tested with all images in the database with 200 images are used in training.

Training and Testing using Inputs from Different Cameras

The proposed techniques are non-intrusive and do not require that the actual camera make/model be used in the training set. To demonstrate this aspect, we test the performance of the proposed techniques using 100 images from Canon Powershot A75 and 100 images from Sony Cybershot DSC P72. We randomly choose 50 out of 100 Canon Powershot A75 images and use them for training to identify the reference pattern; the 100 images from Sony Cybershot DSC P72 are used in testing. The performance results, averaged over 100 iterations, are shown in Figure 4.9. The figure shows that the performance is good for most manipulations and for P_F around 10%, the probability of correct detection is close to $80\% - 90\%$. This result is comparable to the plots in Figure 4.7 and Figure 4.8. The drop in performance for some manipulations such as resampling can be attributed to the absence of the original camera make/model in training.

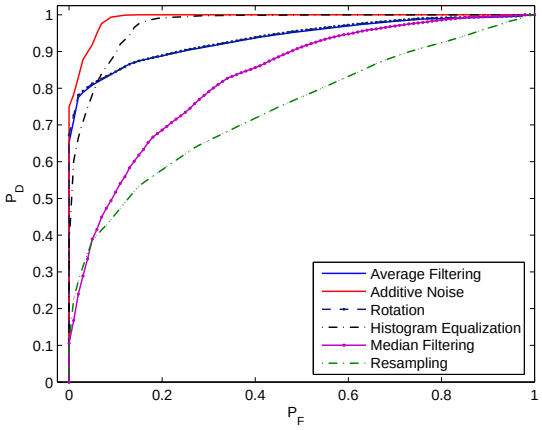


Figure 4.9: Receiver operating characteristics for tampering detection when images from Canon Powershot A75 are used in training and images from Sony Cybershot DSC P72 are used in testing.

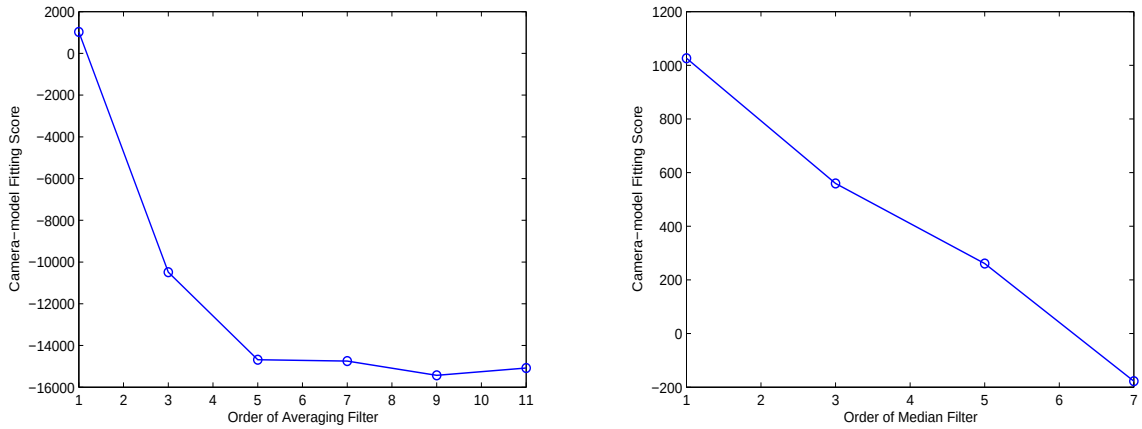
4.3.3 Tampering Forensics using the Estimated Manipulation Filter Coefficients

The estimated filter coefficients can also be employed to quantify the likelihood and degree of tampering, and to identify the type and parameters of the tampering operation. In this subsection, we show that the similarity score can be used to define a *camera-model fitting score* to evaluate the amount of tampering that the test image has undergone. For our experiments, we first choose six good reference patterns that give the highest area under the ROC curve. The camera-model fitting score for the test image is then defined as the median of the similarity scores obtained by comparing the estimated coefficients of the test image with the ones obtained from each of the six reference patterns. The higher the fitting score is, the greater the likelihood that the test image is a direct camera output without further processing.

We examine the variation of the camera-model fitting score as a function of the degree of tampering for all the manipulations listed in Table 4.2. Figure 4.10(a) and Fig 4.10(b) show the camera-model fitting score as a function of the filter order for spatial averaging and median filtering, respectively. In both cases, we observe that the fitting score reduces as the filter order increases and as the degree of tampering increases. Further, the score is less than -1000 for all average filtered images. This low value is because of the distinct nulls in the frequency spectrum of the manipulated filter, estimated from filtered images, making it very different from the flat reference pattern.

Figure 4.11(a) and (b) show the camera-model fitting score as a function of the angle of rotation and the resampling rate, respectively. For manipulations such as rotations, the average fitting scores for manipulated images are less than zero as can be seen in Figure 4.11(a), and therefore the detection algorithm can efficiently identify rotations by setting an appropriate threshold close to zero. For image resampling, the results from Figure 4.11(b) indicate that the average camera-model fitting score reduces as the resampling rate deviates from 100% and therefore these manipulations can be detected with the threshold based classifier. Similar trend is also observed for additive noise and the fitting score reduces as the strength of additive noise increases.

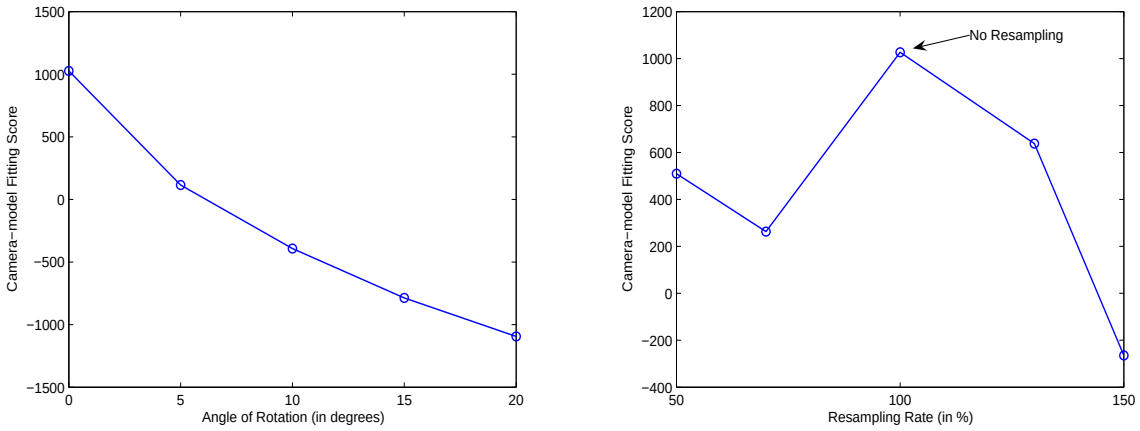
The estimated manipulation filter coefficients can also be employed to identify the type and parameters of post-camera processing operations. In Figure 4.12, we show the frequency response of the estimated manipulation filter coefficients for the different types of manipulations listed in Table 4.2. A closer look at the manipulation filter coefficients in the frequency domain suggest noticeable differences for the different kinds of tampering operations. For such manipulations as



(a) (b)

Figure 4.10: Variation of the camera-model fitting score as a function of the filter order for (a) average filtering and (b) median filtering.

average filtering, we observe distinct nulls in the frequency spectrum and the gap between the nulls can be employed to estimate the order of the averaging filter and its parameters. Image manipulations such as additive noise result in a white noisy spectrum as shown in Figure 4.12(g), and the strength of the noise can be computed from the manipulation filter coefficients. Rotation and downsampling can be identified from the smaller values in the low-high and the high-low bands of the frequency spectrum of the manipulation filter. In our future work, we plan to further investigate on employing the estimated intrinsic fingerprints of post-camera processing operations to provide forensic evidence about the nature and parameters of the tampering that the image has undergone. Such analysis may help re-create the original image from its corresponding tampered versions.



(a) (b)

Figure 4.11: Variation of the camera-model fitting score as a function of the degree of tampering for (a) image rotations and (b) resampling.

4.3.4 Attacking the Proposed Tampering Detection Algorithm

In the work presented so far, we have considered direct camera outputs as authentic images and presented methods to distinguish them from other images that have undergone post-camera manipulations. In this subsection, we examine the other side of the problem from the attackers' viewpoint. Given the knowledge of the proposed tampering detection algorithm, the attacker could potentially come up with better tampering operations to foil the detector. We illustrate it with a particular attack as follows:

In Step 1 of the tampering process, the attackers estimate the color interpolation coefficients using component forensics methodologies described in Section 3.2 . After estimating the color interpolation coefficients, the attacker proceeds to Step 2 to tamper the image by applying such post-camera operations as filtering and resampling; then in Step 3 the attacker re-enforces the camera constraints via

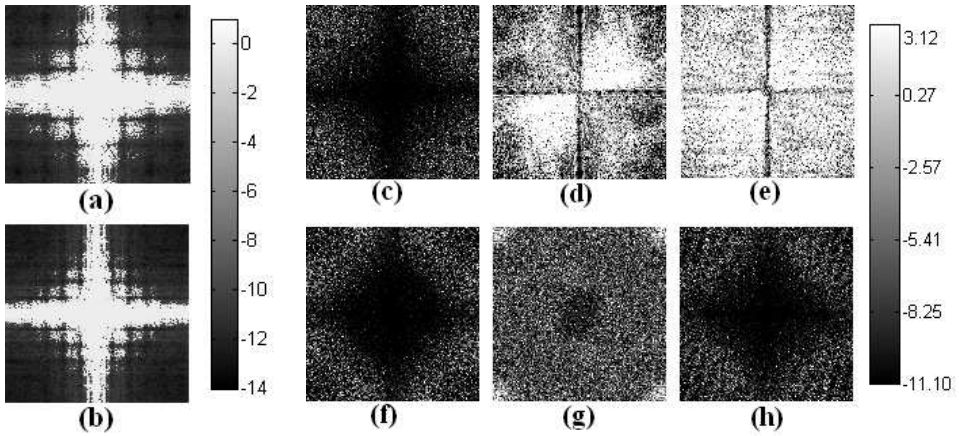
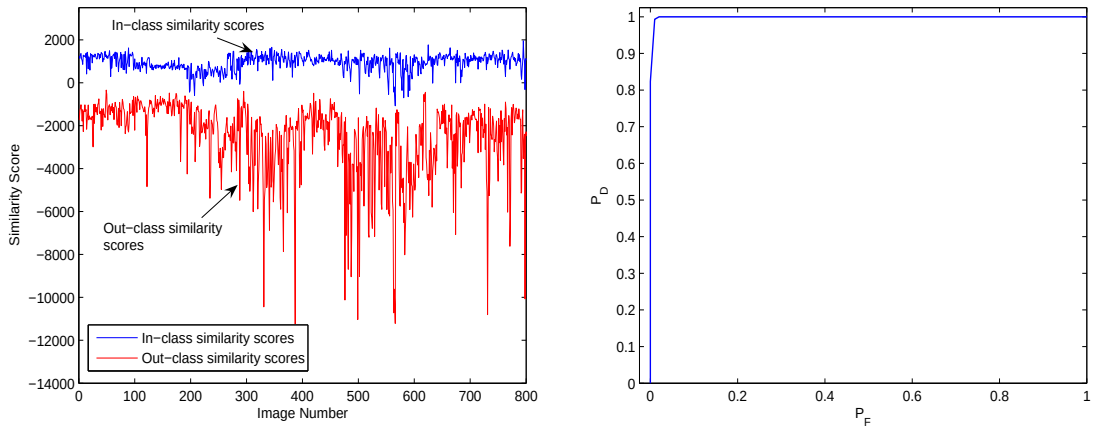


Figure 4.12: Frequency response of the manipulation filter for camera outputs that are manipulated by (a) 7×7 averaging filter, (b) 11×11 averaging filter, (c) 7×7 median filter, (d) 20 degrees rotation, (e) 70% resampling, (f) 130% resampling, (g) noise addition with PSNR 20dB, and (h) histogram equalization. The frequency response is shown in the log scale and shifted so that the DC components are in the center.

(4.3) using the estimated camera component parameters obtained earlier in Step 1.

Figure 4.13(a) shows the in-class and the out-class similarity scores obtained by comparing the reference patterns with the direct camera outputs and the tampered versions by the above three-step process, respectively, for the scenario when the camera input is tampered by down-sampling to half its original size in Step 2, before enforcing the camera constraints in Step 3. We notice from the figure that the in-class and the out-class distances are well separated, and an appropriate threshold value $\tau \approx -200$ can be used to distinguish the two classes. The ROC curve computed using the threshold based classifier is shown alongside in Figure 4.13(b). The figure suggests that the classifier still performs well and gives a P_D close to 100% even for low values of P_F close to 1%. The reason behind the superior performance is because the tampered images have undergone several manipulations,



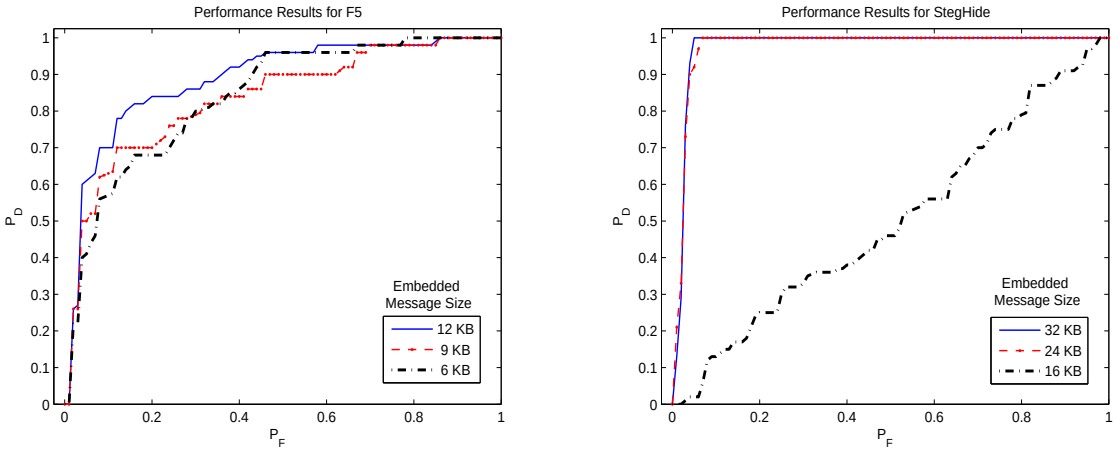
(a) (b)

Figure 4.13: Performance results for reverse engineering attacks: down-sampling by 50% followed by camera-constraint re-enforcement. (a) In-class and out-class similarity scores, (b) Receiver operating characteristics for the tampering detection problem.

each of which introduce some inherent traces in the final output image, and the Step 3 restoration process is not able to completely disguise the attacks from the iterative forensic analysis algorithm. Thus, the proposed techniques can efficiently resist such attacks.

4.4 Further Discussions and Applications

The results in the previous section demonstrate that the intrinsic fingerprint traces left behind in the final digital image by the post-camera processing operations can provide a tell-tale mark to robustly detect global manipulations. In this section, we show that the estimated filter coefficients can also be employed to detect other kinds of post-camera processing operations such as steganographic embedding and watermarking. Further, any change or inconsistencies in the estimated in-camera



(a) (b)

Figure 4.14: Performance results for steganalysis of (a) F5 algorithm and (b) Steghide at different embedding rates.

fingerprints, or the presence of new post-camera fingerprints provide clues to detect cut-paste tampering and to determine if the given image was produced using a camera, a scanner, or a computer graphics software.

4.4.1 Applications to Universal Steganalysis

Watermarking and steganographic embedding may also be modeled as post-processing operations applied to camera outputs, and the estimated post-camera fingerprints can be utilized to identify them. Steganography is the art of secret communication whereby the hidden information is transmitted by embedding it on to the host multimedia. Over the past few years, there have been a number of steganographic embedding algorithms using digital images as hosts for covert communication [44,47,60,91,142]. In the same period, several steganalysis methods have been proposed to identify the presence of hidden data in multimedia. While embedding-specific steganalysis [45] target specific embedding algorithms, universal steganal-

ysis [9, 88] are designed to identify more than one type of steganography. With an increasing number of steganographic embedding algorithms, there is a strong need for robust universal methods for blind steganalysis. As can been seen from our results, the proposed intrinsic fingerprinting techniques facilitate blind steganalysis by distinguishing authentic camera outputs from images with hidden content.

A common challenge of steganalysis is how to model the ground truth original non-stego image data. In our work, we consider direct camera outputs as non-stego data and apply the camera model to characterize its properties; image manipulations such as watermarking and steganography are then modelled as post-processing operations applied to camera outputs. In this subsection, we show that these embedding algorithms leave behind statistical traces on the digital image that can be detected by analyzing the coefficients of the manipulation filter, and examine the performance of our proposed techniques for identifying the presence of hidden messages in multimedia data.

We test the performance of the threshold based detector in distinguishing authentic camera outputs from stego data. In our experiments, we use the same camera data set with 100 images of size 512×512 from Canon Powershot A75 camera [127]. Stego images are then generated by embedding random messages of different sizes into the cover images. Generally speaking, the maximum embedding payload depends on the nature of the cover image and the data hiding algorithm. For our simulations, we first find the average of the maximum embedding payload across 100 images and then embed messages at 100%, 75%, and 50% of this value. For our study, we consider three popular steganographic embedding methods that employ different approaches to hide information – F5 [142], steghide [60], and spread spectrum steganography [91].

Performance Results for LSB Embedding

Least Significant Bit (LSB) embedding methods have been widely used for data hiding. Many algorithms such as Jsteg, JPEG hide-and-seek [77], Outguess [113], and F5 [142] embed a secret message into the LSB of the DCT coefficients of the cover image. For a survey of LSB methods, see [114] and the references therein. Most LSB embedding methods such as JPEG hide-and-seek [77] and Outguess [113] replace the LSB of the DCT coefficients with the secret message, and statistical steganalysis using χ^2-test can be used to detect them [143]. In our work, we focus on the embedding methods of F5 and steghide.

The F5 technique that has been shown to be resilient to such statistical attacks based on χ^2-test [142], although it was subsequently broken in [45] by histogram analysis of DCT coefficients. The F5 embeds data through matrix encoding by decrementing the absolute value of the DCT coefficients. In our experiments with F5, we estimate the average maximum payload across 100 color images to be around 12 KB. The stego images are then generated by embedding secret messages of size 12 KB, 9 KB, and 6 KB using the software [141], respectively. The detection results are shown in Figure 4.14(a) for different embedding rates. We notice that the proposed algorithms perform with reasonable accuracy giving an average detection accuracy close to 62% and 50% respectively at 100% and 75% average embedding rates for false alarm probabilities around 1%. These results are comparable to the wavelet statistics based steganalysis technique [88], which reports average accuracies of 62% and 52% at the embedding rates of 100% and 78%, respectively.

Steghide preserves the first-order statistics of the image and can provide high message capacity. Steghide employs a graph-theoretic approach to embed the

secret messages on multimedia data. The message is hidden by exchanging rather than overwriting pixels [60]. A graph is first constructed from the cover data to the secret message. The pixels to be modified are represented as vertices and are connected to possible partners by edges. A combinatorial problem is then solved to embed the secret message by exchanging samples. In our studies with steghide, we estimate the average maximum payload across 100 color images to be around 32 KB for a 512×512 color image. The stego images are then generated by embedding secret messages of size 32 KB, 24 KB, and 16 KB using the software [59], respectively. The detection results are shown in Figure 4.14(b) for different embedding rates. We notice that the proposed algorithms can efficiently identify steghide at 100% and 75% embedding rates with the probability of identifying stego data close to 100% for a false alarm probability of 1%. However, the performance reduces significantly when the secret message length is reduced to 50% capacity at 16 KB. These results are better than the wavelet statistics based steganalysis technique [88], which reports average accuracies of 77% and 60% at 100% and 78% embedding rates, respectively.

Performance Results for Spread Spectrum Embedding

Next, we study the performance of spread spectrum embedding methods. Block-DCT based spread spectrum embedding have been widely used in literature for data hiding, watermarking, and steganography [146] for a wide variety of applications. Detecting spread spectrum steganography has been a challenging problem over the last decade, and statistics based schemes typically do not perform well in distinguishing original cover data and stego pictures. To our best knowledge, the only work that addresses spread spectrum steganalysis is by Avcibas *et al.* [9],

where it was shown that image quality metrics may be used as features to identify such embedding. In their work, the authors show that they can attain an average probability of correct decision of 80% with 40% false alarm probability when tested with 10 images. We test the performance of the proposed intrinsic fingerprint system for spread spectrum embedding. In our experiments, we use the same camera data set with 100 Canon Powershot A75 images of size 512×512 as our authentic set. Stego images are then generated by adding pseudo-random watermarks at different peak signal-to-noise ratios (PSNR) of 38dB, 40dB, and 42dB. The manipulation filter coefficients are estimated for the cover and the stego data, and classified with the threshold based classifier. Figure 4.15 shows the performance results for different PSNRs. We note that the average identification accuracy is close to 100% for PSNRs of 38dB and 40dB, and reduces to 91% for 42dB PSNR. These results demonstrate the superior performance of the proposed techniques.

In addition to the three steganographic schemes mentioned above, we also test the performance of our algorithms for such embedding techniques as stochastic modulation [44] and perturbed quantization (PQ) steganography [46, 47]. In stochastic modulation steganography [44], a weak noise signal with a noise distribution chosen to mimic the noise produced by the image acquisition device is added to the cover image to embed the message bits. In the case of digital cameras, it has been shown that the sensor and hardware noise are best modelled to be Gaussian distributed [44, 58] and therefore detecting stochastic modulation steganography can be considered equivalent to detecting the presence of additive Gaussian noise in an image captured by a digital camera. Our results suggest that such embedding can be detected with a very high accuracy with a P_D close to 100% for low values of P_F about 1% using the proposed forensic analysis techniques. Perturbed

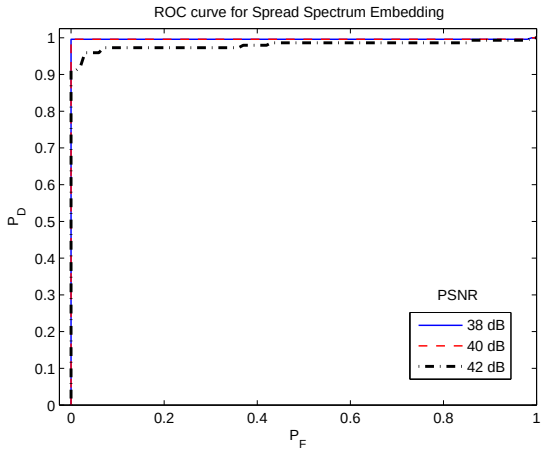


Figure 4.15: Performance results for spread spectrum embedding at different PSNR.

quantization steganography embeds information in the DCT coefficients by quantizing the values either up or down depending upon the message to embedded. The set of changeable coefficients is first found by identifying those coefficients whose fractional part (i.e., difference between the actual value and the quantized value) is lower than a pre-chosen threshold [47]. For our experiments with PQ steganography, we use the 100 Canon Powershot A75 images of size 512×512, JPEG compressed in the camera with the default quality factor close to 97%, as our authentic set. Stego images are created by randomly embedding messages into these images and quantizing them to a quality factor of 70%. Steganalysis for this scheme is more challenging and the proposed techniques are able to identify such manipulations with P_D close to 70-80% under a $P_F \approx 25\%$.

4.4.2 Distinguishing Camera Capture from Other Image Acquisition Processes

The proposed forensics methodology can be used to authenticate the source of the digital color image. Evidence obtained from such forensic analysis would provide useful forensic information to law enforcement and intelligence agencies as to if a given image was actually captured with a camera or scanner, or generated using computer graphics software. We demonstrate this application with two case studies.

Photographs vs Scanned Images

Digital cameras and image scanners are two main categories of image acquisition devices. While a large amount of pictures of natural scenes are taken with digital cameras, scanners have been increasingly used for digitizing documents. Rapid technology development and the availability of high quality scanners has in part led to more sophisticated digital forgeries. In this case study, we are interested in determining if a digital image is produced by a camera or a scanner. The motivation behind employing the proposed techniques for device identification is based on the observation that the manipulation filter coefficients for an authentic camera output would be close to a delta function, and the corresponding coefficients for a scanned image would represent the scan process.

For our study, we choose 25 different images from four camera models to give a total of 100 images for the *camera image* data set. We then collect another set of 25 different photographic images from several cameras with diverse image content. These photographs are printed and then scanned back using 4 different scanner models: (a) Canon CanoScan D1250U2F, (b) Epson Perfection 2450 photo, (c) Mi-

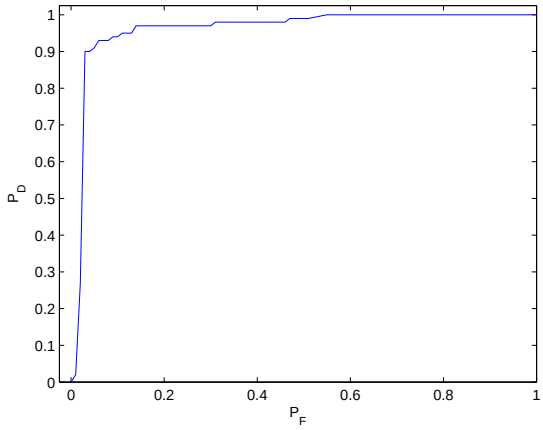


Figure 4.16: Receiver operating characteristics for classifying authentic camera outputs from scanned images.

crotek ScanMaker 3600, and (d) Visioneer OneTouch 5800USB. These $25 \times 4 = 100$ images form our *scanned image* data set. We test our proposed methods for these 200 images. The frequency response of the manipulation filter is estimated and compared with a reference pattern. The ROC obtained using the threshold based classifier is shown in Figure 4.16. Here P_D denotes the fraction of scanned images that are correctly classified as scanned, and P_F represents the fraction of camera outputs mis-classified as scanned. We observe from the figure that the probability of correct decision P_D is around 92% for 1% false probability rate. These results indicate that our proposed methods can effectively distinguish between the camera-captured and scanned images.

Photographs vs Photo Realistic Computer Graphics

With an increasing number of sophisticated processing tools, creating realistic imagery has become easier. Modern graphic synthesis and image rendering tools can be used to reproduce photographs to a very high degree of precision and accuracy,

and therefore, the problem of distinguishing camera outputs from photorealistic computer graphics has become important. In this case study, we employ our proposed framework to distinguish digital photographic images and photorealistic graphics images. For our study, we use a set of 100 images from 4 camera models to create the camera image dataset. A randomly chosen set of 100 photorealistic computer graphics images, obtained from the Columbia dataset [101] constitute our *photorealistic* computer graphics data set. We use a cropped sub-image of size 512×512 to estimate the coefficients of the manipulation filter. The estimated frequency response is then compared with the reference pattern and a threshold based classifier is used to distinguish authentic camera outputs from graphics images. The results of our analysis, in terms of the receiver operating characteristics (ROC), are shown in Figure 4.17. Here P_D denotes the fraction of graphics images that are correctly classified as photorealistic, and P_F represents the fraction of photographs classified as computer generated. A large area under the ROC curve suggests that our proposed method can distinguish between the two classes. These results are comparable to the geometry based features proposed in [102], and are better than the wavelet features [36] and the cartoon features based classifiers tested in [102]. Different from the geometry based features in [102] that are motivated by the modelling the computer graphics creation tools and the artifacts produced therein, our method focuses on finding the algorithms and parameters of the imaging process in digital cameras to distinguish digital photographic images from photorealistic computer graphics.

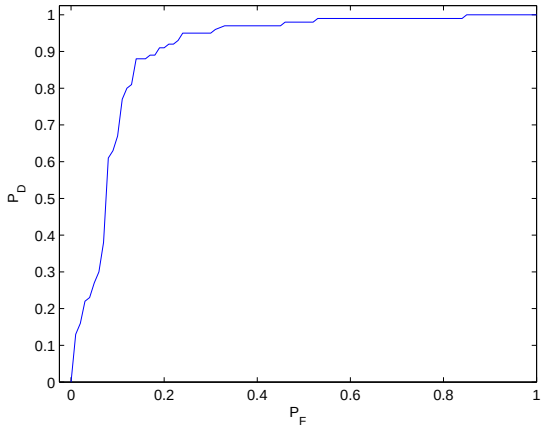


Figure 4.17: Receiver operating characteristics for classifying authentic camera outputs from photorealistic computer graphics.

4.5 Chapter Summary

In this chapter, we propose a set of forensic signal processing techniques to verify whether a given digital image is an direct camera output or not. We introduce a new formulation to study the problem of image authenticity. The proposed formulation is based on the observation that each in-camera and post-camera processing operation leave some distinct intrinsic fingerprint traces on the final image. We characterize the properties of a direct camera output using a camera model, and estimate its component parameters and the intrinsic fingerprints. We consider any further post-camera processing as a manipulation filter, and find the coefficients of its linear shift-invariant approximation using blind deconvolution. A high similarity of the estimated coefficients and the reference pattern that corresponds to no manipulations, certifies the integrity of the given image. We show through detailed simulation results that the proposed techniques can be used to identify different types of post-camera processing, such as filtering, resampling, rotation, etc. Evi-

dence obtained from such forensic analysis is used to build a universal steganalyzer to determine the presence of hidden messages in multimedia data. Our results suggest that we can efficiently detect different types of embedding methods such as least significant bit (LSB) and spread spectrum techniques with a high accuracy. The estimated post-camera fingerprints are also employed for image acquisition forensics to establish if a given digital image is from a digital camera, a scanner, or a computer graphics software. Overall, our proposed techniques provides a common framework for a broad range of forensic analysis on digital images.

Appendix: Convexity of the Optimization Problem and Uniqueness of Solution

In this appendix, we show that the optimization formulation in (4.4) is convex if the camera's color interpolation coefficients are known. A function J is said to be convex if for any u_1, u_2 and $0 \leq \lambda \leq 1$, we have

$$J(\lambda u_1 + (1-\lambda)u_2) \leq \lambda J(u_1) + (1-\lambda)J(u_2).$$

Since $J(u)$ in (4.4) is a sum of two quadratic functions, it is sufficient to show that these two functions are convex. Let

$$J(u) = \sum_{c=1}^{3} (J_c^1(u) + J_c^2(u)),$$

where

$$J_c^1(u) = \sum_{x,y} \left[\sum_{m,n} u(m,n,c) \left(\hat{S}_t(x-m, y-n, c) - S_t(x-m, y-n, c) \right) \right]^2,$$

and

$$J_c^2(u) = (\sum_{m,n} u(m,n,c) - 1)^2.$$

Here, $\hat{S}_t$ denotes the estimate of the test image S_t obtained by imposing the camera constraints:

$$\hat{S}_t(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) S_t(x-m, y-n, c) & \forall \{x,y\} \in \Re_i, \text{ and } 1 \leq c \leq 3, \\ S_t(x,y,c) & \text{otherwise.} \end{cases}$$

In the above equation, $\alpha_{\Re_i}$ denotes the color interpolation coefficients employed in the camera to render the test image S_t. In the absence of additional information, the values of $\alpha_{\Re_i}$ can be non-intrusively estimated from the test image as long as S_t is a direct camera output or an image that has undergone minor levels of post-interpolation processing. Now, defining

$$\varphi_i(x,y,c) = \sum_{m,n} u_i(m,n,c) \left(\hat{S}_t(x-m, y-n, c) - S_t(x-m, y-n, c) \right),$$

we get

$$\begin{aligned} J_c^1(\lambda u_1 + (1-\lambda)u_2) &= \sum_{x,y} \left[\lambda \varphi_1(x,y,c) + (1-\lambda)\varphi_2(x,y,c) \right]^2 \\ &= \lambda \sum_{x,y} \varphi_1(x,y,c)^2 + (1-\lambda) \sum_{x,y} \varphi_2(x,y,c)^2 \\ &\quad -\lambda(1-\lambda) \times \sum_{x,y} \left(\varphi_1(x,y,c) - \varphi_2(x,y,c) \right)^2 \\ &= \lambda J_c^1(u_1) + (1-\lambda) J_c^1(u_2) \\ &\quad -\lambda(1-\lambda) \times \sum_{x,y} \left(\varphi_1(x,y,c) - \varphi_2(x,y,c) \right)^2 \\ &\leq \lambda J_c^1(u_1) + (1-\lambda) J_c^1(u_2), \end{aligned}$$

where the last inequality follows from $0 \leq \lambda \leq 1$. This shows that J_c^1 is convex. Similarly, we can show that the quadratic function J_c^2 is also convex, and therefore establish the convexity of J.

To show that the solution of the optimization problem is unique, we make use of a theorem in optimization theory that states that solution of a convex optimization

problem with a cost function J is unique if the cost function is unimodal [61, 74], *i.e.*, $\nabla^2 J(u) > 0$ for all u. Defining $\Psi(x, y, c) = S_t(x, y, c) - \hat{S}_t(x, y, c)$, we can show that

$$\begin{aligned}\frac{\partial^2 J}{\partial u(a_i, b_i, c)\partial u(a_j, b_j, c)} &= 2\sum_{x,y} \Psi(x - a_i, y - b_i, c)\Psi(x - a_j, y - b_j, c) \\ &\quad +2u(a_i, b_i, c)u(a_j, b_j, c), \\ &= 2 < \Lambda_{(a_i,b_i,c)}, \Lambda_{(a_j,b_j,c)} >,\end{aligned}$$

where $\Lambda_{(a_i,b_i,c)}$ represents a vector of length $(H \times W + 1)$ consisting of all the elements of $\Psi(x - a_i, y - b_i, c)$ for all x and y along with the element $u(a_i, b_i, c)$. Arranging the vectors $\Lambda_{(a_i,b_i,c)}$ column-wise, we construct the matrix $\Omega_c = [\Lambda_{(a_1,b_1,c)}\Lambda_{(a_1,b_2,c)} \ldots]$ of dimension $(H \times W + 1) \times (N_u^2)$ for $c = 1, 2, 3$. We can then show that $\nabla^2 J(u) = 2\sum_{c=1}^{3} \Omega_c \Omega_c^T > 0$. Thus, the cost function is unimodal and therefore its solution unique.

Chapter 5

Theoretical Analysis of Component Forensics

In Chapter 3 and Chapter 4, we introduced *component forensics* as a new methodology for forensic analysis, and showed that evidence obtained from component forensic analysis can be used in a number of applications including discovering patent infringement, authenticating image acquisition source, detecting tampering, and for fostering evolutionary studies. When security is compromised, intellectual rights is violated, or authenticity is forged, component forensic methodologies can be employed to reconstruct what have happened to the content to answer who has done what, when, where, and how. In the previous chapters, we used the *intrinsic fingerprint* traces left behind in the final digital image by the different components of the imaging device as evidence to estimate the component parameters and to answer the forensic questions. However, as the intrinsic fingerprint traces pass through the different parts of the information processing chain, some of them may be modified or destroyed and some others newly created. Therefore, the goodness of this forensic evidence depends to a great extent on the accuracy at which they

can be obtained and this limits their usage.

Let us consider the example of *bootlegging*. In recent times, an increasing number of movies have been re-shot with camcorders directly from the theater where they are screened, and sold in the market. This kind of piracy incurs a significant loss to the copyright industry. Complementary to watermarking and fingerprinting technologies that help track such illegal reproduction, forensic analysis can help to trace the origin and authenticity of digital data. The knowledge of the source camera or camcorder (and its brand/model) that was used to capture the data and information about the the display device (such as a flat-screen or projector) from where the image/video was recorded can help identify both the person who illegally captured the video and the place where the video was shot. To establish such forensic evidence regarding the source and display characteristics in courts, a higher confidence in the decision and a higher accuracy in parameter estimation is strongly desired. However, such accuracies may not always be attained in practice via multimedia forensic analysis due to its inherent *fundamental limits*. In the bootlegging example, some traces of the projector employed in the theater might be lost and new fingerprint traces about the camcorder itself might be inserted. Hence, the data obtained from the final camcorder alone may or may not help compute the parameters of the display device. This leads to further forensic questions as to what components are identifiable and what are not.

In this chapter, we introduce a novel theoretical framework for component forensics to quantify the accuracies at which the intrinsic fingerprints and the component parameters can be estimated. We develop formal notions of identifiability of components and investigate fundamental performance bounds. We define a *component* as the basic unit of the information processing chain to facilitate

theoretical analysis and consider two different scenarios. In the first scenario, we assume no prior knowledge about the component or the possible subset of algorithms employed by the component, and develop a framework based on estimation theory and Cramer-Rao lower bounds to quantify the accuracies in estimating the parameters of several components in the information processing chain [126, 134]. Details of this work are presented in Section 5.1. This theoretical framework has useful in applications where there the forensic analyst has no prior knowledge about the forensic system.

In some forensic applications, additional side information may be available to the forensic analyst [130,134]. For instance, in the bootlegging example, geographic constraints can be enforced to narrow down on a possible set of theaters (and their display parameters) from where the movie could have been illegally recorded using a camcorder. In the presence of such additional information, the component parameters could be found with a higher accuracy from among the available sample set of algorithms by reformulating the estimation problem as a classification problem. In Section 5.2, we consider this scenario and develop a theoretical framework for media forensics under the assumption that the component parameters take values from a finite set of possible algorithms. We derive conditions under which a component is *forensically classifiable* and present case studies to demonstrate the applications of this framework for a wide range of forensic tasks.

5.1 Theoretical Analysis via Estimation Framework

In this section, we introduce a theoretical framework for component forensics and examine the conditions under which the parameters of a component can be estimated accurately. We quantify the accuracy of estimation in terms of *bias* and *variance* of the estimator and derive performance bounds based on Fisher Information. We first review Fisher information in Section 5.1.1 and then introduce the theoretical formulation in Section 5.1.2.

5.1.1 Fisher Information and Cramer-Rao Lower Bound

Fisher information is the amount of information that an observable random variable Z carries about an unobservable parameter θ [48]. It is mathematically given by

$$\mathcal{I}(Z,\theta) = E_\theta \left\{ \left[\frac{\partial}{\partial \theta} \ln f(Z|\theta) \right]^2 \right\}, \tag{5.1}$$

where $f(Z|\theta)$ denotes the probability density function (pdf) of Z conditioned on the value of the parameter to be estimated θ, and the notation E_θ denotes that the expectation is performed conditioned on the value of the parameter θ. The significance of the Fisher information is given by the *Cramer-Rao* lower bound (CRLB). According to the CRLB, the average estimation error given an estimator $\hat{\theta}(Z)$ is lower bounded by

$$E_\theta(\hat{\theta}(Z) - \theta)^2 \geq \frac{\left[1 + \frac{\partial}{\partial \theta} b(\hat{\theta},\theta)\right]^2}{E_\theta \left\{ \left[\frac{\partial}{\partial \theta} \ln f(Z|\theta) \right]^2 \right\}} + b(\hat{\theta},\theta)^2, \tag{5.2}$$

where $b(\hat{\theta},\theta)$ denotes the *bias* of the estimator and is given by

$$b(\hat{\theta},\theta) = E_\theta(\hat{\theta}(Z)) - \theta. \tag{5.3}$$

If the estimator, $\hat{\theta}(Z)$, is unbiased, $b(\hat{\theta}, \theta) = 0$ and (5.2) reduces to

$$E_\theta(\hat{\theta}(Z) - \theta)^2 \geq \frac{1}{E_\theta\left\{\left[\frac{\partial}{\partial\theta}\ln f(Z|\theta)\right]^2\right\}} = \mathcal{I}(Z,\theta)^{-1}, \tag{5.4}$$

suggesting that the variance of the estimator is lower bounded by the inverse of Fisher information.

5.1.2 Theoretical Analysis using Fisher Information: Background and Definitions

To facilitate theoretical analysis, let $\Re_x$ denote a super-set of all possible inputs that can be given to the k^{th} component $\mathcal{C}_k$, and let $\Re_y$ contain the corresponding outputs. Without loss of generality, let $x \in \Re_x$ be the input and $y \in \Re_y$ denote the corresponding output. Now, we have the following definitions:

Definition 5.1 *The parameter θ_k of a component $\mathcal{C}_k$ can be estimated **intrusively** using an estimator $\hat{\theta}_k(y,x)$ with an average error $E_{\theta_k}(\hat{\theta}_k(y,x) - \theta_k)^2$ such that*

$$E_{\theta_k}(\hat{\theta}_k(y,x) - \theta_k)^2 \geq \frac{\left[1 + \frac{\partial}{\partial\theta_k}b(\hat{\theta}_k,\theta_k)\right]^2}{E_{\theta_k}\left\{\left[\frac{\partial}{\partial\theta_k}\ln f(y|x,\theta_k)\right]^2\right\}} + b(\hat{\theta}_k,\theta_k)^2 = \delta_k(x). \tag{5.5}$$

where $b(\hat{\theta}_k,\theta_k)$ denotes the bias term given by

$$b(\hat{\theta}_k,\theta_k) = E_\phi(\hat{\theta}_k(y,x)) - \theta_k. \tag{5.6}$$

From the CRLB, it can be shown that any other estimator $\mathcal{T}(y,x)$ of the parameter θ_k cannot provide error values lower than $\delta_k(x)$, *i.e.*,

$$E_\theta\left((\mathcal{T}(y,x) - \theta_k)|x\right)^2 \geq \delta_k(x). \tag{5.7}$$

If the forensic analyst is not allowed to break open the device, then he/she can either do semi non-intrusive or completely non-intrusive analysis depending on the

availability of the device. In this case, we may extend the definition to study multi-component devices. Let a device $\mathcal{D}$ with N_c components be represented as $\mathcal{D} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{N_c}\}$, and let $\phi = [\theta_1, \theta_2, \ldots, \theta_{N_c}]^T$ denote set of the parameters of all the N_c components in the device. We may now define the following:

Definition 5.2 *The parameter set ϕ of the device $\mathcal{D}$ can be estimated **semi non-intrusively** with an average error $E_\phi\left[(\hat{\phi}(y,x)-\phi)(\hat{\phi}(y,x)-\phi)^T\right]$ using an estimator*

$$\hat{\phi}(y,x) = [\hat{\theta_1}(y,x), \hat{\theta}_2(y,x), \ldots, \hat{\theta_{N_c}}(y,x)]^T,$$

of the parameter set ϕ, such that $E_\phi\left[(\hat{\phi}(y,x)-\phi)(\hat{\phi}(y,x)-\phi)^T\right] \geq \Delta_s(x)$ where

$$\Delta_s(x) = \left(\frac{\partial}{\partial \phi^T}\mathbf{b}_s(\hat{\phi},\phi)\right)\mathcal{I}_s(x,\phi)^{-1}\left(\frac{\partial}{\partial \phi^T}\mathbf{b}_s(\hat{\phi},\phi)\right)^T + \mathbf{b}_s(\hat{\phi},\phi)\mathbf{b}_s(\hat{\phi},\phi)^T. \quad (5.8)$$

Here,

$$\mathbf{b}_s(\hat{\phi},\phi) = E_\phi(\hat{\phi}(y,x)) - \phi, \quad (5.9)$$

represents the bias term, and $\mathcal{I}_s(x,\phi)$ denotes the Fisher information matrix for semi non-intrusive forensics with its $(i,j)^{\text{th}}$ element given by

$$\mathcal{I}_s^{ij}(x,\phi) = E_\phi\left[\frac{d}{d\theta_i}\ln f(y|x,\phi)\frac{d}{d\theta_j}\ln f(y|x,\phi)\right]. \quad (5.10)$$

As can be seen from (5.8), the accuracy of parameter estimation depends on the choice of the input to the system and can be improved by designing better inputs. Motivated by this observation, we define a notion of an *optimal* input as follows:

Definition 5.3 *An **optimal** input for semi non-intrusive forensics, $\hat{x}_e$, is the one that minimizes the average error in parameter estimation,* i.e.,

$$\hat{x}_e = \arg\min_{x\in\Re_x} ||\Delta_s(x)||_d, \quad (5.11)$$

where $||\Delta_s(x)||_d$ *represents an appropriate matrix norm or a function of* $\Delta_s(x)$. *The lowest error that can achieved via semi non-intrusive analysis is then given by* $||\Delta_s||_d = ||\Delta_s(\hat{x})||_d$.

Several definitions of $||.||_d$ are possible. Unless otherwise specified, in this work, we define minimum of $||.||_d$ to represent element-wise minima. More specifically, for two matrices Δ_1 and Δ_2, we say $\Delta_1 < \Delta_2$ if all the elements of Δ_1 are less than the corresponding entries of Δ_2; and based on this definition, find the optimal input as the one that minimizes all the array elements. This definition of $||.||_d$ could be restrictive in certain applications as there might not be one single input that minimizes all the entries of the matrix. Later in this section, we consider particular examples for which this might be possible.

In the case of *non-intrusive* forensics, the forensic analyst does not have access to the camera at hand and only has some sample images provided to him. Therefore, in this case, the estimate is done without the knowledge of the input x.

Definition 5.4 *A device* $\mathcal{D}$ *with parameter set* ϕ *can be estimated* ***non-intrusively*** *with an average error* $E_\phi\left[(\hat{\phi}(y)-\phi)(\hat{\phi}(y)-\phi)^T\right] \geq \Delta_n$ *using the estimator,* $\hat{\phi}(y) = [\hat{\theta_1}(y), \hat{\theta_2}(y), \ldots, \hat{\theta_{N_c}}(y)]^T$, *of the parameter set* ϕ, *where*

$$\Delta_n = \left(\frac{\partial}{\partial \phi^T}\mathbf{b}_n(\hat{\phi},\phi)\right)\mathcal{I}_n(\phi)^{-1}\left(\frac{\partial}{\partial \phi^T}\mathbf{b}_n(\hat{\phi},\phi)\right)^T + \mathbf{b}_n(\hat{\phi},\phi)\mathbf{b}_n(\hat{\phi},\phi)^T. \quad (5.12)$$

Here, the bias term is given by

$$\mathbf{b}_n(\hat{\phi},\phi) = E_\phi(\hat{\phi}(y)) - \phi, \quad (5.13)$$

and $\mathcal{I}_n(\phi)$ *denotes the Fisher information matrix for completely non-intrusive forensics with its* $(i,j)^{\text{th}}$ *element given by*

$$\mathcal{I}_n^{ij}(\phi) = E_\phi\left[\frac{d}{d\theta_i}\ln f(y|\phi)\frac{d}{d\theta_j}\ln f(y|\phi)\right]. \quad (5.14)$$

If the estimator is unbiased, the bias term corresponding to $\mathbf{b}_s(\hat{\phi}, \phi)$ in (5.9) and $\mathbf{b}_n(\hat{\phi}, \phi)$ in (5.13) are zero and therefore the error terms $\Delta_s(x)$ and Δ_n depend only on the Fisher information as

$$\Delta_s(x) = \mathcal{I}_s(x, \phi)^{-1}, \tag{5.15}$$

$$\Delta_n = \mathcal{I}_n(\phi)^{-1}. \tag{5.16}$$

Here, the $(i, i)^{\text{th}}$ of the matrix $\Delta_s(x)$ denotes the error in estimating the parameter of the i^{th} component, and the $(i, j)^{\text{th}}$ cross terms of the matrix represent the interaction between the components i and j in the device.

For the case of digital cameras, the best estimates for most components such as color interpolation and white balancing are typically unbiased in nature. Therefore, in the rest of our work, we assume an unbiased estimator for which the best achievable accuracies under semi non-intrusive and completely non-intrusive scenarios are given by (5.15) and (5.16), respectively.

5.1.3 Theoretical Analysis and Fundamental Limits

We may now theoretically establish the following results. All the results presented in this section are for unbiased estimators.

Theorem 5.1 *The average Fisher information obtained for component parameter estimation via semi non-intrusive forensics is larger than the the corresponding Fisher information for completely non-intrusive forensics. Expressed mathematically,*

$$E(\mathcal{I}_s(x, \phi)) \geq \mathcal{I}_n(\phi), \tag{5.17}$$

where the expectation is performed over all x in the input space $\Re_x$.

Proof: We show the proof for a single component system and the analysis can be extended to devices with multiple components. For a device with one component with parameter $\phi = \theta$, we have

$$\mathcal{I}_s(x,\phi) = E_\phi\left\{\left[\frac{\partial}{\partial\phi}\ln f(y|x,\phi)\right]^2\right\}, \quad \text{and} \tag{5.18}$$

$$\mathcal{I}_n(\phi) = E_\phi\left\{\left[\frac{\partial}{\partial\phi}\ln f(y|\phi)\right]^2\right\}. \tag{5.19}$$

Here, the expectations are performed over all output values y given the input x and component parameter ϕ in (5.18) and over all output values y given the the component parameter ϕ in (5.19), respectively. Taking expectation with respect to x on both sides of (5.18), we have

$$\begin{aligned} E(\mathcal{I}_s(x,\phi)) &= E\left(E_\phi\left\{\left[\frac{\partial}{\partial\phi}\ln f(y|x,\phi)\right]^2\right\}\right), \\ &= \int_{x\in\Re_x}\int_{y\in\Re_y}\left[\frac{\partial}{\partial\phi}\ln f(y|x,\phi)\right]^2 f(y|x,\phi)p(x)dydx, \\ &= \int_{y\in\Re_y}\left[\int_{x\in\Re_x}\left(\frac{\partial}{\partial\phi}f(y|x,\phi)\right)^2\frac{1}{f(y|x,\phi)}p(x)dx\right]dy. \end{aligned} \tag{5.20}$$

Writing (5.19) as

$$\mathcal{I}_n(\phi) = \int_{y\in Re_y}\left(\frac{\partial}{\partial\phi}f(y|\phi)\right)^2\frac{1}{f(y|\phi)}dy, \tag{5.21}$$

and expanding $f(y|\phi)$, we get

$$\mathcal{I}_n(\phi) = \int_{y\in\Re_y}\left(\int_{x\in\Re_x}\frac{\left\{\frac{\partial}{\partial\phi}f(y|x,\phi)\right\}}{\sqrt{f(y|x,\phi)}}\sqrt{p(x)}\times\sqrt{p(x)f(y|x,\phi)}dx\right)^2\frac{1}{f(y|\phi)}dy. \tag{5.22}$$

Applying *Cauchy-Schwarz inequality* to the above equation gives

$$\begin{aligned}
\mathcal{I}_n(\phi) &\leq \int_{y\in\Re_y}\left(\int_{x\in\Re_x}\frac{\left\{\frac{\partial}{\partial\phi}f(y|x,\phi)\right\}^2}{f(y|x,\phi)}p(x)dx\right)\times\left(\int_{x\in\Re_x}p(x)f(y|x,\phi)dx\right)\frac{1}{f(y|\phi)}dy,\\
&= E(\mathcal{I}_s(x,\phi))\times\left(\int_{x\in\Re_x}p(x)f(y|x,\phi)dx\right)\frac{1}{f(y|\phi)}dy,\\
&= E(\mathcal{I}_s(x,\phi)). \qquad (5.23)
\end{aligned}$$

This completes the proof of the theorem. ∎

Theorem 5.1 suggests, as a corollary, that for an unbiased estimator, the component parameter estimation errors obtained using non-intrusive forensics are greater than the average error obtained via semi non-intrusive analysis, or $\text{diag}\{E(\Delta_s(x))\} \leq \text{diag}\{\Delta_n\}$. Here, '$\leq$' of two matrices represents element-wise comparison. Semi non-intrusive forensics provides additional control to the forensic analyst both in terms of designing device inputs and input conditions to give better component parameter estimation results, and this intuitively justifies the reason behind the result that even on an average, the performance of semi non-intrusive forensics would be better than completely non-intrusive forensics.

Corollary 5.1 *The Fisher information obtained for component parameter estimation via semi non-intrusive forensics is larger than the the corresponding Fisher information for completely non-intrusive forensics.* i.e., $\mathcal{I}_s(\phi) = \mathcal{I}_s(\hat{x}_e,\phi) \geq \mathcal{I}_n(\phi)$.

Proof: The proof of the corollary follows from Theorem 5.1 where we showed that $E(\Delta_s(x)) \leq \Delta_n$, where '$\leq$' represents element-wise inequality. By definition of optimal input for semi non-intrusive forensics, we have

$$\Delta_s = \min_{x\in\Re_x}\Delta_s(x) \leq E(\Delta_s(x)) \leq \Delta_n.$$

This suggests that $\Delta_s \leq \Delta_n$ which completes the proof. ∎

This result suggests that for an unbiased estimator, the component parameter estimation errors obtained via semi non-intrusive analysis would be lower than that obtained via completely non-intrusive analysis, and semi non-intrusive forensics is therefore better. Next, we examine the conditions under which both semi non-intrusive and completely non-intrusive analysis give the same accuracies.

Theorem 5.2 *The Fisher information obtained for component parameter estimation via semi non-intrusive forensics is equal to the Fisher information for completely non-intrusive forensics when the knowledge of the component parameters do not help in the guessing the input x given the output y. In this scenario, semi non-intrusive forensics and completely non-intrusive analysis provides the same accuracies.*

Proof: From the definition of Fisher information for semi non-intrusive forensics, we have

$$\begin{aligned}
\mathcal{I}_s^{ij}(x,\phi) &= E_\phi\left\{\left[\frac{\partial}{\partial\theta_i}\ln\left(\frac{f(x|y,\phi)f(y|\phi)}{p(x)}\right)\right]\times\left[\frac{\partial}{\partial\theta_j}\ln\left(\frac{f(x|y,\phi)f(y|\phi)}{p(x)}\right)\right]\right\}, \\
&= \mathcal{I}_n^{ij}(\phi) + E_\phi\left\{\frac{\partial}{\partial\theta_i}\ln(f(x|y,\phi))\frac{\partial}{\partial\theta_j}\ln(f(x|y,\phi))\right\} \\
&\quad + E_\phi\left\{\frac{\partial}{\partial\theta_i}\ln(f(x|y,\phi))\frac{\partial}{\partial\theta_j}\ln(f(y|\phi))\right\} \\
&\quad + E_\phi\left\{\frac{\partial}{\partial\theta_j}\ln(f(x|y,\phi))\frac{\partial}{\partial\theta_i}\ln(f(y|\phi))\right\}, \qquad (5.24)
\end{aligned}$$

A closer look at (5.24) shows that the equality $\mathcal{I}_s(x,\phi) = \mathcal{I}_n(\phi)$ is attained when $\forall i$, $\frac{\partial}{\partial\theta_i}\ln(f(x|y,\phi)) = 0$, or $f(x|y,\phi)$ is independent of the component parameters θ_i for all $1 \le i \le N_c$. This result also suggests that the knowledge of the component parameters do not help in the guessing the input x given the output y; thus, completing the proof of the theorem. ∎

Now, let us consider an example for illustration.

Example: Consider a device with a single component for which the input-output relationship is given by

$$y = \alpha x + n,$$

where x represents the input to the component, y denotes the corresponding output, α is a constant, and n represents additive noise. For this example, let us assume that n follows a Gaussian distribution with mean 0 and variance Σ_n. $\phi = \{\alpha, \Sigma_n\}$ is the component parameter set.

Using the definition of Fisher information for semi non-intrusive forensics, we can show that

$$\mathcal{I}_s(x, \phi) = \begin{bmatrix} \frac{x^2}{\Sigma_n} & 0 \\ 0 & \frac{3}{4\Sigma_n^2} \end{bmatrix}. \tag{5.25}$$

As a first step, we compute the optimal input for semi non-intrusive forensics from (5.25). We observe from the equation that optimal input maximizes $\mathcal{I}_s^{11}(x, \phi) = \frac{x^2}{\Sigma_n}$, which is the signal-to-noise ratio (SNR) with signal power equal to x^2 and noise power given by the variance of the noise signal Σ_n. This suggests that the optimal input for semi non-intrusive forensics of this component would be the input that maximizes $||x||_d$. Defining $||x||_d$ to be the norm of x, we find the optimal input as the one that maximizes the signal power, *i.e.*, $\hat{x}_e = max_{x \in \Re_x} |x|$.

Next, we derive the Fisher information for completely non-intrusive forensics under the premise that the input to the system follows a Gaussian distribution with mean μ_x and variance Σ_x, *i.e.*, $x \sim \mathcal{N}(\mu_x, \Sigma_x)$. With this assumption, it can be shown that the output y also follows a Gaussian distribution with $y \sim \mathcal{N}(\alpha\mu_x, \alpha^2\Sigma_x + \Sigma_n)$ and therefore, we have

$$\mathcal{I}_n(\phi) = \begin{bmatrix} \frac{3\alpha^2\Sigma_x^2}{(\alpha^2\Sigma_x+\Sigma_n)^2} + \frac{\mu_x^2}{(\alpha^2\Sigma_x+\Sigma_n)} & \frac{3\alpha\Sigma_x}{2(\alpha^2\Sigma_x+\Sigma_n)^2} \\ \frac{3\alpha\Sigma_x}{2(\alpha^2\Sigma_x+\Sigma_n)^2} & \frac{3}{4(\alpha^2\Sigma_x+\Sigma_n)^2} \end{bmatrix}. \tag{5.26}$$

Now, we use the example as an illustration to verify the above mentioned theorems. Taking expectations on both sides of (5.25) under the assumption that $x \sim \mathcal{N}(\mu_x, \Sigma_x)$, we have

$$E(\mathcal{I}_s(x,\phi)) = \begin{bmatrix} \frac{\Sigma_x+\mu_x^2}{\Sigma_n} & 0 \\ 0 & \frac{3}{4\Sigma_n^2} \end{bmatrix}. \tag{5.27}$$

Taking the term-by-term difference of (5.26) and (5.27), we get

$$E(\mathcal{I}_s^{11}(x,\phi)) - \mathcal{I}_n^{11}(\phi) = \frac{\Sigma_x\left((\alpha^2\Sigma_x - \Sigma_n)^2 + \alpha^2\Sigma_x\Sigma_n\right)}{\Sigma_n(\alpha^2\Sigma_x+\Sigma_n)^2} + \frac{\mu_x^2\alpha^2\Sigma_x}{\Sigma_n(\alpha^2\Sigma_x+\Sigma_n)}, \tag{5.28}$$

$$E(\mathcal{I}_s^{22}(x,\phi)) - \mathcal{I}_n^{22}(\phi) = \frac{3\alpha^2\Sigma_x}{4\Sigma_n^2}\left[\frac{\alpha^2\Sigma_x + 2\Sigma_n}{(\alpha^2\Sigma_x+\Sigma_n)^2}\right]. \tag{5.29}$$

Both these terms satisfy $E(\mathcal{I}_s^{ii}(x,\phi)) - \mathcal{I}_n^{ii}(\phi) \geq 0$ for $i \in \{1,2\}$, verifying Theorem 5.1. This result suggests that on an average semi non-intrusive forensics can provide higher estimation accuracies and lower estimation errors than completely non-intrusive forensics. Further, it can be seen from (5.28) and (5.29) that the condition $E(\mathcal{I}_s^{ii}(x,\phi)) - \mathcal{I}_n^{ii}(\phi) = 0$ is satisfied only when $\Sigma_x = 0$ or x is deterministic with a value equal to μ_x. This confirms the result in Theorem 5.2 indicating that semi non-intrusive forensics and completely non-intrusive forensics can provide the same accuracies when the knowledge of the component parameters do not help in the guessing the input x given the output y. ∎

Theorem 5.3 *For an unbiased estimator, the component parameter estimation errors obtained via intrusive analysis is lower than or equal to the average estimation errors obtained using semi non-intrusive studies.*

Proof: We first consider a simple case of a device $\mathcal{D}$ consisting of two components namely, $\mathcal{C}_1$ and $\mathcal{C}_2$ and prove the theorem for this case. Let x be the input to the

device and $x \in \Re_x$, and let y be the output of the device, $y \in \Re_y$. For sake of analysis, we define a variable z as the output of the first component which is provided as an input to the second. Also, let $\Re_z$ denote the superset of all possible intermediate outputs so that $z \in \Re_z$. Let $\phi = [\theta_1 \ \theta_2]^T$ denote the device parameter set with θ_1 and θ_2 representing the parameters of the first and the second component, respectively. The first component $\mathcal{C}_1$ with parameter θ_1 takes an input x and outputs a value z with probability $p_{\theta_1}(z|x)$; and the second component takes z as the input and outputs $y \in \Re_y$ with probability $q_{\theta_2}(y|z)$. The overall input-output relationship is then given by $\mathcal{P}(y|x)$ where

$$\mathcal{P}_\phi(y|x) = \int_{z\in\Re_z} p_{\theta_1}(z|x) q_{\theta_2}(y|z) dz. \tag{5.30}$$

The $(1,1)^{\text{th}}$ term of the Fisher information corresponding to semi non-intrusive forensics, $\mathcal{I}_s^{11}(x)$ can be written as

$$\begin{aligned}
\mathcal{I}_s^{11}(x) &= \int_{y\in\Re_y} \left(\frac{\partial}{\partial\theta_1}\mathcal{P}_\phi(y|x)\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)} dy \\
&= \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)\right\} q_{\theta_2}(y|z) dz\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)} dy \\
&= \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)}{\sqrt{p_{\theta_1}(z|x)}} \times \sqrt{q_{\theta_2}(y|z)}\right\} \times \sqrt{p_{\theta_1}(z|x) q_{\theta_2}(y|z)} dz\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)} dy \\
&\leq \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)}{\sqrt{p_{\theta_1}(z|x)}}\right\}^2 q_{\theta_2}(y|z) dz\right) \\
&\qquad\qquad \times \left(\int_{z\in\Re_z} p_{\theta_1}(z|x) q_{\theta_2}(y|z) dz\right) \frac{1}{\mathcal{P}_\phi(y|x)} dy \\
&= \int_{z\in\Re_z} \left\{\frac{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)}{\sqrt{p_{\theta_1}(z|x)}}\right\}^2 \times \left(\int_{y\in\Re_y} q_{\theta_2}(y|z) dy\right) dz = \mathcal{I}_i^{11}(x).
\end{aligned} \tag{5.31}$$

Therefore, we have $\mathcal{I}_s^{11}(x) \leq \mathcal{I}_i^{11}(x)$ for all $x \in \Re_x$. Denoting the optimal inputs for semi non-intrusive forensics and intrusive forensics by $\hat{x}_e^{\text{semi}}$ and $\hat{x}_e^{\text{int}}$, respectively, we have $\mathcal{I}_s^{11}(\hat{x}_e^{\text{semi}}) \leq \mathcal{I}_i^{11}(\hat{x}_e^{\text{semi}}) \leq \mathcal{I}_i^{11}(\hat{x}_e^{\text{int}}) = \max_{x\in\Re_x} \mathcal{I}_i^{11}(x)$. Similarly, it

can be shown for all $x \in \Re_x$ that

$$\begin{aligned}
\mathcal{I}_s^{22}(x) &= \int_{y\in\Re_y} \left(\frac{\partial}{\partial\theta_2}\mathcal{P}_\phi(y|x)\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)}dy \\
&= \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\partial}{\partial\theta_2} q_{\theta_2}(y|z)\right\} p_{\theta_1}(z|x)dz\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)}dy \\
&= \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\frac{\partial}{\partial\theta_2} q_{\theta_2}(y|z)}{\sqrt{q_{\theta_2}(y|z)}} \times \sqrt{p_{\theta_1}(z|x)}\right\} \times \sqrt{p_{\theta_1}(z|x)q_{\theta_2}(y|z)}dz\right)^2 \frac{1}{\mathcal{P}_\phi(y|x)}dy \\
&\leq \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{\frac{\frac{\partial}{\partial\theta_2} q_{\theta_2}(y|z)}{\sqrt{q_{\theta_2}(y|z)}}\right\}^2 p_{\theta_1}(z|x)dz\right) \\
&\qquad\qquad \times \left(\int_{z\in\Re_z} p_{\theta_1}(z|x)q_{\theta_2}(y|z)dz\right) \frac{1}{\mathcal{P}_\phi(y|x)}dy \\
&= \int_{z\in\Re_z} p_{\theta_1}(z|x)dz \int_{y\in\Re_y} \left\{\frac{\frac{\partial}{\partial\theta_1} q_{\theta_2}(y|z)}{\sqrt{q_{\theta_2}(y|z)}}\right\}^2 dy = \int_{z\in\Re_z} p_{\theta_1}(z|x)\mathcal{I}_i^{22}(z)dz \\
&= E(I_i^{22}(z)|x) \leq \max_{z\in\Re_z} I_i^{22}(z) = \mathcal{I}_i^{22}(\hat{x}_e^{\text{int}}). \qquad (5.32)
\end{aligned}$$

Therefore, we have specifically for $x = \hat{x}_e^{\text{semi}}$ that $\mathcal{I}_s^{22}(\hat{x}_e^{\text{semi}}) \leq \mathcal{I}_i^{22}(\hat{x}_e^{\text{int}})$.

This result suggests that the diagonal elements of the Fisher information matrix satisfy $\text{diag}(\mathcal{I}_s(\hat{x}_e^{\text{semi}})) \leq \text{diag}(\mathcal{I}_i(\hat{x}_e^{\text{int}}))$. Therefore, for an unbiased estimator, the component parameter estimation errors obtained using semi non-intrusive forensics are greater than the average error obtained via intrusive analysis, or $\text{diag}\{E(\Delta_i(x))\} \leq \text{diag}\{\Delta_s\}$. This proves the theorem. ∎

In the case of semi non-intrusive forensics the decision has to be made based on the overall input-output response of the entire device. Therefore the final forensic analysis is this case is dependent upon how different components in the device interact with each other and to what extent the intrinsic fingerprint traces of one component are lost/modified when they pass through the other components in the information processing chain. This reduces the overall accuracies of semi non-intrusive forensics. On the other hand, in the case of intrusive analysis, the

forensic analyst can break open the device and examine each and every component individually independent of the other components in the information processing chain. In the following theorem, we mathematically derive the conditions under which semi non-intrusive analysis can provide the same accuracies as intrusive analysis.

Theorem 5.4 *For an unbiased estimator, the component parameter estimation errors obtained via intrusive analysis is equal to the average estimation errors obtained using semi non-intrusive studies only if the mutual Fisher information between any two components in the system is equal to zero.*

Proof: To prove this theorem, we take a closer look at the estimation errors to examine the conditions under which semi non-intrusive analysis gives the same accuracies compared to intrusive analysis. For an unbiased estimator, the average estimation errors obtained from intrusive analysis for a two component device with parameter set ϕ are given by

$$\Delta_i(x) = \begin{bmatrix} \delta_i^{11}(x) & \delta_i^{12}(x) \\ \delta_i^{21}(x) & \delta_i^{22}(x) \end{bmatrix} = \mathcal{I}_i(x,\phi)^{-1} = \begin{bmatrix} 1/\mathcal{I}_i^{11}(x,\theta_1) & 0 \\ 0 & 1/\mathcal{I}_i^{22}(x,\theta_2) \end{bmatrix}, \quad (5.33)$$

$$\begin{aligned} \Delta_s(x) = \begin{bmatrix} \delta_s^{11}(x) & \delta_s^{12}(x) \\ \delta_s^{21}(x) & \delta_s^{22}(x) \end{bmatrix} &= \mathcal{I}_s(x,\phi)^{-1} = \frac{1}{|\mathcal{I}_s(x,\phi)|} \begin{bmatrix} \mathcal{I}_s^{22}(x,\phi) & -\mathcal{I}_s^{21}(x,\phi) \\ -\mathcal{I}_s^{12}(x,\phi) & \mathcal{I}_s^{11}(x,\phi) \end{bmatrix}, \\ &= \begin{bmatrix} \frac{1}{\mathcal{I}_s^{11}(x,\theta_1) - \frac{(\mathcal{I}_s^{12}(x,\phi))^2}{\mathcal{I}_s^{22}(x,\theta_2)}} & -\frac{\mathcal{I}_s^{12}(x,\phi)}{|\mathcal{I}_s(x,\phi)|} \\ -\frac{\mathcal{I}_s^{12}(x,\phi)}{|\mathcal{I}_s(x,\phi)|} & \frac{1}{\mathcal{I}_s^{22}(x,\phi) - \frac{(\mathcal{I}_s^{12}(x,\phi))^2}{\mathcal{I}_s^{11}(x,\phi)}} \end{bmatrix}, \end{aligned} \quad (5.34)$$

where the last equation follows from the fact that $\mathcal{I}_s^{12}(x,\phi) = \mathcal{I}_s^{21}(x,\phi)$. Moreover, as the magnitude of $\mathcal{I}_s^{12}(x,\phi)$ increases, the estimation error increases. Comparing the two equations, we notice that the equality is attained only when the following

conditions are satisfied

$$\mathcal{I}_s^{11}(x,\phi) = \mathcal{I}_i^{11}(x,\theta_1), \tag{5.35}$$

$$\mathcal{I}_s^{22}(x,\phi) = \mathcal{I}_i^{22}(x,\theta_2), \quad \text{and} \tag{5.36}$$

$$\mathcal{I}_s^{12}(x,\phi) = 0. \tag{5.37}$$

It is to be noted that the $\mathcal{I}_s^{12}(x,\phi)$ term represents the interactions between the two components and higher its absolute value, the greater the interaction. A small absolute value of $\mathcal{I}_s^{12}(x,\phi)$ suggests that the components are independent and its parameters can be estimated separately. Further, from the inequalities in (5.31), we notice that the condition $I_s^{11}(x) = I_i^{11}(x)$ is satisfied only when

$$\begin{aligned} \forall z, \quad & \frac{\left\{ \frac{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)}{\sqrt{p_{\theta_1}(z|x)}} \sqrt{q_{\theta_2}(y|z)} \right\}}{\sqrt{p_{\theta_1}(z|x) q_{\theta_2}(y|z)}} = \text{constant} \\ \text{or} \quad \forall z, \quad & \frac{\frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x)}{p_{\theta_1}(z|x)} = \text{constant(say } \mathrm{c}_1), \end{aligned} \tag{5.38}$$

and the inequalities in (5.32) becomes an equality only when

$$\forall z, \quad \frac{\frac{\partial}{\partial\theta_2} q_{\theta_2}(y|z)}{q_{\theta_2}(y|z)} = \text{constant(say } \mathrm{c}_2). \tag{5.39}$$

Expanding on $\mathcal{I}_s^{12}(x,\phi)$, we have

$$\begin{aligned} \mathcal{I}_s^{12}(x,\phi) &= \int_{y\in\Re_y} \left(\frac{\partial}{\partial\theta_1} \mathcal{P}_\phi(y|x) \right) \times \left(\frac{\partial}{\partial\theta_2} \mathcal{P}_\phi(y|x) \right) \frac{1}{\mathcal{P}_\phi(y|x)} dy \\ &= \int_{y\in\Re_y} \left(\int_{z\in\Re_z} \left\{ \frac{\partial}{\partial\theta_1} p_{\theta_1}(z|x) \right\} q_{\theta_2}(y|z) dz \right) \\ &\qquad \times \left(\int_{z\in\Re_z} \left\{ \frac{\partial}{\partial\theta_2} q_{\theta_2}(y|z) \right\} p_{\theta_1}(z|x) dz \right) \frac{1}{\mathcal{P}_\phi(y|x)} dy. \end{aligned} \tag{5.40}$$

Substituting for (5.38) and (5.39), we have

$$\mathcal{I}_s^{12}(x,\phi) = \int_{y\in\Re_y} c_1 c_2 \mathcal{P}_\phi(y|x) dy = c_1 c_2 \tag{5.41}$$

Therefore, $\Delta_i(x) = \Delta_s(x)$ would be satisfied only when $c_1 c_2 = 0$ or either of c_1 or c_2 is zero. This suggests that either $p_{\theta_1}(z|x)$ is independent of the parameter θ_1 or $q_{\theta_2}(y|z)$ is independent of θ_2 or mathematically

$$\frac{\partial}{\partial \theta_1} p_{\theta_1}(z|x) = 0, \quad \text{or } \frac{\partial}{\partial \theta_2} q_{\theta_2}(y|z) = 0. \tag{5.42}$$

The above equations will be satisfied only when $\mathcal{I}_s^{ij}(x, \phi) = 0$. This completes the proof of the theorem. ∎

This theorem leads to the following definition:

Definition 5.5 *Two components of the device are said to be* ***forensically independent*** *if its component parameters can be estimated separately and the errors in estimating the parameters of one component does not affect the estimation of the other components' parameters.*

As can be seen from the previous theorem, two components would be forensically independent if and only if $\mathcal{I}_s^{ij}(x, \phi) = 0$. *For a device with more than two components, this condition reduces to* $\mathcal{I}_s^{ij}(x, \phi) = 0$ *and* $\forall k \quad \mathcal{I}_s^{ik}(x, \phi) = 0$ *or* $\mathcal{I}_s^{kj}(x, \phi) = 0$.

Corollary 5.2 *Intrusive analysis, semi non-intrusive forensic analysis, and completely non-intrusive forensic analysis provide the same accuracies in parameter estimation only when all the components in the device are forensically independent of each other. i.e.,* $\mathcal{I}_s^{ij}(x, \phi) = 0$ *for all* i *and* j *such that* $i \neq j$ *and* $1 \leq i, j \leq N_c$.

Proof: The proof of this corollary follows from the proofs of Theorem 5.2 and Theorem 5.4. ∎

Next, we consider an example of forensically independent components and to illustrate how this theoretical analysis can be employed to compute optimal inputs,

and later in Chapter 6, we employ these principles to design optimal inputs for semi non-intrusive forensics of digital cameras to identify color interpolation and white balancing components.

Example: Consider a system with the input-output response given by

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{5.43}$$

where $\underline{x} = [x_1 \ x_2]^T$ is the input to the system, $\underline{y} = [y_1 \ y_2]^T$ denotes the output from the system, $\underline{n} = [n_1 \ n_2]^T$ represents additive Gaussian noise with $E(n_1^2) = E(n_2^2) = \Sigma_n$ and $E(n_1 n_2) = 0$, and $\phi = [a_{11} \ a_{12} \ a_{21} \ a_{22}]^T$ are the component parameters. The goal of the forensic analyst is to compute the values of the parameter ϕ.

In this example of a single component system, the Fisher information matrix under semi non-intrusive forensics can be shown to be given by

$$\mathcal{I}_s(x, \phi) = \frac{4}{\Sigma_n} \begin{bmatrix} x_1^2 & x_1 x_2 & 0 & 0 \\ x_1 x_2 & x_2^2 & 0 & 0 \\ 0 & 0 & x_1^2 & x_1 x_2 \\ 0 & 0 & x_1 x_2 & x_2^2 \end{bmatrix}. \tag{5.44}$$

From the matrix, we observe the following:

- A higher value of x_1 and x_2 can provide higher accuracies in parameter estimation. This is because a higher value would imply that the signal power is much larger than the noise power giving a higher SNR.

- The estimation of the component parameters a_{11} and a_{12} are dependent on each other because of the non-zero value of $\mathcal{I}_s^{12}(x, \phi)$ for non-zero inputs. This observation can be intuitively explained by the fact that the estimation

of both the component parameters a_{11} and a_{12} needs to be done based on the same equation

$$y_1 = a_{11}x_1 + a_{12}x_2 + n_1. \tag{5.45}$$

Additionally, we notice that $\mathcal{I}_s^{12}(x, \phi) = 0$ only when $x_1 = 0$ or $x_2 = 0$ in which case (5.45) reduces to either $y_1 = a_{11}x_1 + n_1$ or $y_1 = a_{12}x_2 + n_1$. Under these conditions, the component parameters a_{11} and a_{12} can be estimated independent of each other from one of the two reduced equations.

- The estimation of the component parameters a_{11} and a_{21} (or a_{22}) are independent of each other as can be seen from the Fisher information matrix ($\mathcal{I}_s^{13}(x, \phi) = \mathcal{I}_s^{14}(x, \phi) = 0$). This is because a_{11} is solely estimated from (5.45) and the equation $y_2 = a_{21}x_1 + a_{22}x_2 + n_2$ does not provide any information to aid in the estimation of a_{11}.

Based on these observations, we can conclude that there is no single optimal input for semi non-intrusive forensics. The best strategy for the forensic analyst would be to first give an input $\underline{x}$ with $x_1 = \max_{x \in \Re_x} x$ and $x_2 = 0$ and observe the output to estimate the values of the parameters a_{11} and a_{21}, and then give the input $\underline{x}$ with $x_1 = 0$ and $x_2 = \max_{x \in \Re_x} x$ to obtain a_{12} and a_{22}. In this way, the analyst can design good inputs to improve the overall accuracy of parameter estimation. ∎

5.2 Theoretical Analysis via Pattern Classification Framework

In this section, we develop a theoretical framework for media forensics for components with a finite number of possibilities in the parameter space. The proposed

framework employs ideas from pattern classification theory to answer forensic questions about what components and processing operations are classifiable and what are not. We define formal notions of identifiability of components under different scenarios, and quantify the confidence in which the component parameters can be computed in each case. The analysis presented in this section adds to the understanding of multimedia forensics and supplements the the theoretical analysis based on estimation theory presented in the previous section. We that the confidence in identifying the component parameters depends on the nature of available inputs and testing conditions, and that intrusive forensics gives higher confidence than semi non-intrusive forensics and semi non-intrusive analysis is better than completely non-intrusive scenario.

5.2.1 Background and Definitions

As in Section 5.1, we consider a system with N_c components $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{N_c}\}$ and let $\Re_x$ and $\Re_y$ denote the set of all possible inputs and outputs respectively. Unlike in the previous case where we assume that the parameter of the k^{th} component θ^k can take infinite number of possibilities, in this section, we develop a new theoretical framework under the premise that the component parameter can take a finite number of values from the algorithm space, *i.e.,* $\theta^k \in \Theta^k = \{\theta_1^k, \theta_2^k, \ldots, \theta_{N_a}^k\}$, where N_a^k is the total number of possible algorithms for the component $\mathcal{C}_k$. Now, we define formal notions of intrusively, semi non-intrusively, and completely non-intrusively classifiable components.

Definition 5.6 *A component $\mathcal{C}_k$ is said to be* ***intrusively classifiable*** *or* ***i-classifiable*** *if for each possible algorithm θ_i^k used by the component, and for most*

inputs $x \in \Re_x$,

$$p(\theta_i^k|y,x) \geq p(\theta_j^k|y,x) \quad \forall j \in \{1,2,\ldots,N_a^k\} \text{ and } j \neq i,$$

and there exists at least one input $x^* \in \Re_x$ *and its corresponding output* y^* *for which*

$$p(\theta_i^k|y^*,x^*) > p(\theta_j^k|y^*,x^*) \; \forall j \in \{1,2,\ldots,N_a^k\} \text{ and } j \neq i.$$

Here, x *and* y *denote the corresponding input and output of the component, respectively, and are vectors of appropriate dimensions; and* p *denotes the probability distribution function. The forensic analyst can then employ maximum a posteriori estimation techniques [31] to identify the component parameters* $\hat{\theta}^k$ *as*

$$\hat{\theta}^k = \arg \max_{j=1,2,\ldots,N_a^k} p(\theta_j^k|y,x).$$

In semi non-intrusive and completely non-intrusive forensics, analysts are not allowed to break open the device or system. In the scenario of *semi non-intrusive* forensics, the analysts have access to the system as a black box, and can design appropriate inputs to the system and collect the corresponding output data in order to analyze the processing techniques and compute the parameters of the individual components. To examine this scenario, we define $\phi_j = [\theta_{j_1}^1, \theta_{j_2}^2, \ldots, \theta_{j_{N_c}}^{N_c}]$ to represent the set of algorithms (and parameters) employed by the entire system. Assuming that the component parameters in the k^{th} component can take N_a^k possibilities, we have a total of $N_a = \prod_{k=1}^{N_c} N_a^k$ possible algorithm choices for the system. The task for the forensic analyst is now reduced to finding which of these N_a algorithms is used by the system in question.

Definition 5.7 *A system is said to be* ***semi non-intrusively classifiable*** *or* ***s-classifiable*** *if for each possible algorithm* ϕ_i *used by the component, and for most*

inputs $x \in \Re_x$

$$p(\phi_i|y,x) \geq p(\phi_j|y,x) \quad \forall j \in \{1,2,\ldots,N_a\} \text{ and } j \neq i, \tag{5.46}$$

and there exists at least one input $x^* \in \Re_x$ *and its corresponding output* y^* *such that*

$$p(\phi_i|y^*,x^*) > p(\phi_j|y^*,x^*) \; \forall j \in \{1,2,\ldots,N_a\} \text{ and } j \neq i. \tag{5.47}$$

Here, x *and* y *denote the inputs and its corresponding outputs, respectively, of the overall system.*

In addition to computing the parameters of the internal building blocks of the components, it is also important to know the confidence level on the parameter estimation result. A higher confidence value would increase the trustworthiness of the decision made by the forensic analyst in applications involving infringement/licensing to determine potential technology breach [128, 130]; and also in cases involving tampering detection.

Definition 5.8 *For an* ***s-classifiable*** *system with parameter set* ϕ_i*, the confidence score* $\eta_i^{(\text{semi})}(x,y)$ *for correct classification under the input* x *and its corresponding output* y *is defined by the difference between the likelihood of the correct decision and the average of the corresponding likelihoods of the making a wrong decision. Expressed mathematically,*

$$\begin{aligned}
\eta_i^{(\text{semi})}(x,y) &= p(\phi_i|y,x) - \frac{1}{N_a-1}\sum_{j=1,j\neq i}^{N_a} p(\phi_j|y,x), \\
&= p(\phi_i|y,x) - \frac{1}{N_a-1}(1-p(\phi_i|y,x)), \\
&= \frac{N_a}{N_a-1}\left(p(\phi_i|y,x) - \frac{1}{N_a}\right).
\end{aligned} \tag{5.48}$$

As can be seen from the equation, the confidence score $\eta_i^{(\text{semi})}(x, y)$ is proportional to the difference between the probability of correct classification and $(1/N_a)$ that corresponds to uniform likelihood. In our work, we define the confidence score using (5.48) motivated by Definition 5.6 and 5.7. Several other definitions for the confidence score in classification have been proposed in literature [104,128,130,139]. Later in Section 5.2.3, we examine other definitions of confidence score.

The equation (5.48) also suggests that the confidence score is a function of the input x and can be improved by selecting proper inputs. To illustrate this aspect of confidence score, we consider the following example.

Example: Consider an example of a component with parameters $\{\xi_0, \xi_1\}$ whose input-output relationship is given by:

$$y(n) = \xi_0 x(n) + \xi_1 x(n-1).$$

Let $x^{(1)} = [\ldots, 1, 1, 1, \ldots]$ and $x^{(2)} = [\ldots, 0, 1, 2, \ldots]$ be two possible inputs to the system. The corresponding outputs would be $y^{(1)} = [\ldots, \xi_0+\xi_1, \xi_0+\xi_1, \xi_0+\xi_1, \ldots]$ and $y^{(2)} = [\ldots, -\xi_1, \xi_0, 2\xi_0+\xi_1, \ldots]$, respectively. We notice that $y^{(1)}$ is a constant sequence with each of its elements being equal to $(\xi_0 + \xi_1)$ and knowledge of the sum would not provide any indicative of the parameters ξ_0 or ξ_1. Therefore, $x^{(1)}$ is not a good input for evaluating the value of the component. On the other hand, observing the output $y^{(2)}$ of the system, one can formulate a system of linear equations to compute the value of ξ_0 and ξ_1; thus, $x^{(2)}$ is a good input to obtain the component parameter values.

More generally, let us define $\mathbf{q}(x, y) = [p(\phi_1|y, x),\ p(\phi_2|y, x),\ \ldots, p(\phi_{N_a}|y, x)]$ to facilitate discussions. If for an input, x', $\mathbf{q}(x', y') = [0,\ \ldots, 1,\ 0,\ \ldots, 0]$ with 1 at the i^{th} location, the decision of choosing the i^{th} class is made with a very

high confidence and $\eta_i^{(\text{semi})}(x', y')$ equal to 1. On the other hand, if $\mathbf{q}(x'', y'') = [\frac{1-\varepsilon}{N_a}, \ldots, \frac{1}{N_a} + \frac{N_a-1}{N_a}\varepsilon, \ \frac{1-\varepsilon}{N_a}, \ \ldots, \frac{1-\varepsilon}{N_a}]$ where ε is a small positive real number, there is an almost equal probability that the given data sample comes from any of the N_a classes. In this case, the decision is made with a very low confidence with $\eta_i^{(\text{semi})}(x'', y'') = \varepsilon \approx 0$. In this example, x' and x'' represent the best and the worst possible inputs for identifying the component parameters. For other inputs, x, the value of $\eta_i^{(\text{semi})}(x, y)$ would lie in the interval [0, 1], with a higher value indicating more confidence in the decision made. ∎

This example illustrates that the confidence score in parameter estimation can be improved by choice of inputs, and generalizing on this observation, we define an *optimal input* as the one that maximizes the confidence score [130].

Definition 5.9 *An **optimal input**, $\hat{x}_i$, for semi non-intrusive forensic analysis of the system that employs the algorithm ϕ_i is defined as the one that maximizes the confidence score, i.e.,*

$$\hat{x}_i = \arg\max_{x \in \Re_x} \eta_i^{(\text{semi})}(x, y). \tag{5.49}$$

*The corresponding confidence score, $\eta_i^{(\text{semi})} = \eta_i^{(\text{semi})}(\hat{x}_i, \hat{y}_i)$, then represents the overall maximum confidence in **semi non-intrusively** classifying the parameters of the system, where $\hat{y}_i$ is the output of the system with input $\hat{x}_i$.*

In the *completely non-intrusive* forensics scenario, the forensic analyst is provided only with some sample data produced by the device or system and does not have access to nor other knowledge about its inputs. In this case, we can define:

Definition 5.10 *A system is said to be completely **non-intrusively classifiable** or **n-classifiable** if for each possible algorithm ϕ_i used by the component, and*

most possible outputs $y \in \Re_y$,

$$p(\phi_i|y) \geq p(\phi_j|y) \quad \forall j \in \{1, 2, \ldots, N_a\} \text{ and } j \neq i, \tag{5.50}$$

and there exists at least one input $x^* \in \Re_x$, *such that the corresponding output,* y^*, *satisfies*

$$p(\phi_i|y^*) > p(\phi_j|y^*) \quad \forall j \in \{1, 2, \ldots, N_a\} \text{ and } j \neq i. \tag{5.51}$$

The confidence score for a system to be non-intrusively classifiable under the output y *when the actual algorithm employed is* ϕ_i *is given by*

$$\eta_i^{(\text{non})}(y) = \frac{N_a}{N_a - 1}\left(p(\phi_i|y, x) - \frac{1}{N_a}\right). \tag{5.52}$$

5.2.2 Major Results

We now establish the following results.

Theorem 5.5 *If a system is* ***n-classifiable****, then it is* ***s-classifiable****.*

Proof: If a device is *n-classifiable*, then for each possible algorithm $\phi_i (1 \leq i \leq N_a)$ used by the component, there exists an input $x \in \Re_x$ to the overall system such that its corresponding output y satisfies

$$p(\phi_i|y) > p(\phi_j|y) \text{ for } j = 1, 2, \ldots, N_a, j \neq i, \tag{5.53}$$

$$\int_{\Re_x} p(\phi_i|y, x)p(x)dx > \int_{\Re_x} p(\phi_j|y, x)p(x)dx \text{ for } j = 1, \ldots, N_a, j \neq i. \tag{5.54}$$

Since, all the terms on both sides of the equation are positive, there must be atleast one $x = x_0 \in \Re_x$ for which

$$p(\phi_i|y, x_0)p(x_0) > p(\phi_j|y, x_0)p(x_0) \text{ for } j = 1, 2, \ldots, N, j \neq i. \tag{5.55}$$

Factoring out $p(x_0)$ completes the proof. ∎

Theorem 5.6 *The confidence scores obtained using semi non-intrusive analysis is greater than or equal to the ones obtained via completely non-intrusive analysis. i.e., If a system is n-classifiable with a confidence score* $\eta_i^{(\text{non})}(y)$ *under the output* y*, then it is s-classifiable with a confidence score* $\eta_i^{(\text{semi})} \geq \eta_i^{(\text{non})}(y)$.

Proof: From the definition of the confidence score for semi non-intrusive forensics for the input x, we have

$$\eta_i^{(\text{semi})}(x,y) = \frac{N_a}{N_a - 1}\left(p(\phi_i|y,x) - \frac{1}{N_a}\right). \tag{5.56}$$

Multiplying the equations with $p(x)$ and integrating over $\Re_x$, we obtain:

$$\begin{aligned} E(\eta_i^{(\text{semi})}(x,y)) &= \int_{x\in\Re_x} \eta_i^{(\text{semi})}(x,y)p(x)dx \\ &= \int_{x\in\Re_x} \left\{\frac{N_a}{N_a - 1}\left(p(\phi_i|y,x) - \frac{1}{N_a}\right)\right\} p(x)dx \\ &= \frac{N_a}{N_a - 1}\left(p(\phi_i|y) - \frac{1}{N_a}\right) = \eta_i^{(\text{non})}(y). \end{aligned} \tag{5.57}$$

Thus, we have $\eta_i^{(\text{semi})} = \max_{x\in\Re_x} \eta_i^{(\text{semi})}(x,y) \geq E(\eta_i^{(\text{semi})}(x,y)) = \eta_i^{(\text{non})}(y)$; this completes the proof of the theorem. ∎

Theorem 5.5 and Theorem 5.6 suggest that if a component is non-intrusively classifiable, then its parameters can also be identified semi non-intrusively, and the average confidence values obtained using semi non-intrusive analysis is greater than or equal to the ones obtained via completely non-intrusive analysis under a given output. These results pertain to the scenario where the forensic analyst has to make a decision based on 'one' output or 'one' input-output pair. If the forensic analyst has access to 'multiple' outputs or 'multiple' input-output pairs, he/she can then make a combined judgement based on studying all the available data samples. In the following, we extend the proposed theoretical framework to address such scenarios. We begin with the following lemma.

Lemma 5.1 *The overall confidence in estimating the component parameter(s) given N_d inputs (and corresponding outputs) is lower than the value obtained for the best input/output pair.*

Proof: Suppose $\{y_1, y_2, \ldots, y_{N_d}\}$ denote the N_d output data samples available to the forensic analyst, and let $\{x_1, x_2, \ldots, x_{N_d}\}$ be the corresponding inputs. Then, for a given algorithm ϕ_i, the confidence in parameter estimation is given by

$$\eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}) = \frac{N_a}{N_a - 1}\left(p(\phi_i | x_1, y_1, x_2, y_2, \ldots, x_{N_d}, y_{N_d}) - \frac{1}{N_a}\right). \tag{5.58}$$

Expanding the equation using the independence property, we get

$$\eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}) = \frac{N_a}{N_a - 1}\left(\prod_{m=1}^{N_d} p(\phi_i | x_m, y_m) - \frac{1}{N_a}\right). \tag{5.59}$$

Now, let $\hat{m} = \arg\max_{m=1,2,\ldots,N_d} p(\phi_i | x_m, y_m)$ so that $p(\phi_i | x_{\hat{m}}, y_{\hat{m}}) \geq p(\phi_i | x_m, y_m)$ for all $m \in \{1, 2, \ldots, N_d\}$. Equation (5.59) can therefore be re-written in terms of $p(\phi_i | x_{\hat{m}}, y_{\hat{m}})$ to give

$$\begin{aligned} \eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}) &\leq \frac{N_a}{N_a - 1}\left(p(\phi_i | x_{\hat{m}}, y_{\hat{m}})^{N_d} - \frac{1}{N_a}\right) \\ &\leq \eta_i^{(\text{semi})}(x_{\hat{m}}, y_{\hat{m}}). \end{aligned} \tag{5.60}$$

Thus, we have $\eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}) \leq \max_{m=1,2,\ldots,N_d} \eta_i^{(\text{semi})}(x_m, y_m)$. This completes the proof. ∎

Lemma 5.1 suggests that the highest confidence in parameter estimation is determined by the best input – one among the N_d inputs that gives the maximum confidence score. The remaining inputs would reduce the confidence score and confuse the forensic analyst into possibly making a wrong decision. This result is useful to study the scenario of completely non-intrusive forensics. In this

case, the forensic analyst does not have access to the device at hand and collects the forensic evidence based on the observed output data available to him. More specifically, if the analyst has access to N_d such outputs, the overall confidence in his decision can be shown from Theorem 5.6 to be upper bounded by $\eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d})$, *i.e.*,

$$\eta_i^{(\text{non})} \leq \eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}). \tag{5.61}$$

Additionally, the result from Lemma 5.1 gives

$$\eta_i^{(\text{semi})}(x_1, x_2, \ldots, x_{N_d}, y_1, \ldots, y_{N_d}) \leq \max_{m=1,2,\ldots,n} \eta_i^{(\text{semi})}(x_m, y_m) \leq \eta_i^{(\text{semi})}(\hat{x}, \hat{y}) = \eta_i^{(\text{semi})}. \tag{5.62}$$

where $\hat{x}$ denotes the *optimal* input for semi non-intrusive forensics of $\mathcal{D}$. Combining (5.61) and (5.62), we obtain $\eta_i^{(\text{non})} \leq \eta_i^{(\text{semi})}$. This result leads to the following theorem:

Theorem 5.7 *The confidence scores obtained using semi non-intrusive analysis under the optimal input is greater than or equal to the ones obtained via completely non-intrusive analysis even when completely non-intrusive forensics is performed with infinite amount of data.*

Proof: The proof follows from (5.61) and (5.62). ∎

This result is intuitively expected from the fact that semi non-intrusive forensics provides more control to the forensic analyst who can design better inputs to improve the overall performance. Next, we examine the scenario when semi non-intrusive forensics and completely non-intrusive forensics provides the same confidence.

Theorem 5.8 *The confidence scores for component parameter estimation via semi non-intrusive forensics is equal to the confidence scores for completely non-intrusive forensics when the knowledge of the component parameters do not help in the guessing the input x given the output y. In this scenario, semi non-intrusive forensics and completely non-intrusive analysis provides the same accuracies.*

Proof: From the definitions of confidence scores for semi and completely non-intrusive forensics in (5.48) and (5.52), we can show that $\eta_i^{(\text{non})} = \eta_i^{(\text{semi})}$ when $p(\phi_i|y) = p(\phi_i|y,x), \forall i$. It can be shown that this condition is equivalent to $p(x|y,\phi_i) = p(y,x)/p(x)$ or $p(x|y,\phi_i)$ is independent of the component parameters ϕ_i for all $1 \leq i \leq N_a$. This result also suggests that the knowledge of the component parameters do not help in the guessing the input x given the output y; thus, completing the proof of the theorem. ∎

It is to be noted that Theorem 5.8 provides the same conditions for equality of semi and completely non-intrusive forensics as Theorem 5.2 discussed in Section 5.1 and proved via estimation theory. While the theoretical results obtained via estimation and pattern classification theories are based on different assumptions, applicable for different scenarios, and are derived using different mathematical premises, they provide the same fundamental results. This suggests that these theories are merely two different approaches to look at the same problem.

In the remainder of this subsection, we examine the relations between semi non-intrusive forensics and intrusive forensics.

Theorem 5.9 *If a device is **s-classifiable**, then each of its components are **i-classifiable**.*

Proof: This theorem is straightforward if the device has only one component. In this case, the definitions of *s-identifiability* and *i-identifiability* coincide.

Now, let us consider a multi-component device. Let x^k represent the individual inputs for the k^{th} component $\mathcal{C}_k$ (and outputs of the $(k-1)^{\text{th}}$ component), with $x^1 = x$. Since the device is *s-classifiable*, there exists at least one input $x \in \Re_x$ to the overall system such that its corresponding output y satisfies

$$p(\phi_i|y,x) > p(\phi_j|y,x) \text{ for } j = 1,2,\ldots,N, j \neq i, \tag{5.63}$$

for each possible algorithm $\phi_i (1 \leq i \leq N_a)$ used by the component. Writing ϕ_i as $\phi_i = [\theta^1_{i_1}, \theta^2_{i_2}, \ldots, \theta^{N_c}_{i_{N_c}}]$ and expanding $p(\phi_i|y,x)$, we have

$$\begin{aligned} p(\phi_i|y,x) &= p(\theta^1_{i_1}, \theta^2_{i_2}, \ldots, \theta^{N_c}_{i_{N_c}}|y,x) = \prod_{m=1}^{N_c} p(\theta^m_{i_m}|y,x), \\ &= \left(\prod_{m=1}^{N_c} p(\theta^m_{i_m}|x^{m+1},x^m)\right) \prod_{m=1}^{N_c} p(x^{m+1}|x^m,y)p(x^m|y,x). \end{aligned} \tag{5.64}$$

For (5.63) to hold for all $j \neq i$, each of the individual terms in the right hand side of the (5.64) need to satisfy $\forall m \in \{1,2,\ldots,N_c\}$

$$p(\theta^m_{i_m}|x^{m+1},x^m) > p(\theta^m_{j_m}|x^{m+1},x^m) \text{ for all } i_m \neq j_m \text{ and } 1 \leq i_m, j_m \leq N^m_a, \tag{5.65}$$

otherwise, we can construct another hypothesis ϕ_l by replacing the component parameter setting for some of the components. This contradicts (5.63) as there exists atleast one $j = l$ for which $p(\phi_i|y,x) \leq p(\phi_j|y,x)$. Equation (5.65) also shows the existence of alteast one input input $x = x^j$ to the j^{th} component for which the component would be *i-classifiable*. This completes the proof of the theorem. ∎

In general, the converse of Theorem 5.9 is not true. To examine the conditions under which an *i-classifiable* component is *s-classifiable*, we introduce the notion of an ϵ-*consistent* component.

Definition 5.11 *A component is said to be* ϵ-***consistent*** *if the following two conditions are satisfied:*

1. *for most outputs y_1 and y_2 with $d_Y(y_1, y_2) \leq \epsilon$, the estimates of the corresponding inputs x_1 and x_2 satisfy $d_X(x_1, x_2) \leq \epsilon$, where d_X and d_Y are appropriately chosen distance metrics in the input and the output space, respectively,*

2. *for most inputs x_1 and x_2 with $d_X(x_1, x_2) \leq \epsilon$, the estimates of the corresponding outputs y_1 and y_2 satisfy $d_Y(y_1, y_2) \leq \epsilon$.*

We now have the following theorem that relates the confidence in intrusively classifying a component and the confidence values obtained for semi non-intrusively classifying the same component.

Theorem 5.10 *If all the components in a system are ϵ-consistent and the k^{th} component with parameter θ_i^k is **i-classifiable** with a confidence score $\eta_i^{k(\text{int})}$, then the k^{th} component is **s-classifiable** with confidence score $\eta_i^{k(\text{semi})}$ approximately given by*

$$\eta_i^{k(\text{semi})} \approx \eta_i^{k(\text{int})} - 2(N_c - 1)\epsilon \left| \frac{\partial \eta_i^{k(\text{int})}(x, y)}{\partial x} \right|_{x=\hat{x}_i^k, y=\hat{y}_i^k}. \tag{5.66}$$

Proof: In the ideal case, highest confidence $\eta_i^{k(int)}$ is attained when the input to the k^{th} component is the optimal input denoted as $\hat{x}_i^k$ (with its corresponding output $\hat{y}_i^k$). However, since the $(k-1)$ prior to $\mathcal{C}_k$ are ϵ-consistent, it would not be possible to exactly attain $\hat{x}_i^k$, but only $(k-1) \times (2\epsilon)$ close to it. This would lead to a confidence drop of $(k-1) \times (2\epsilon) \left| \frac{\partial \eta_i^{k(\text{int})}(x,y)}{\partial x} \right|_{x=\hat{x}_i^k, y=\hat{y}_i^k}$. Since, the forensic analyst can only observe the final output y, he/she would incur an additional error of from the remaining $(N_c - k)$ components equal to $(N_c - k) \times (2\epsilon) \left| \frac{\partial \eta_i^{k(\text{int})}(x,y)}{\partial x} \right|_{x=\hat{x}_i^k, y=\hat{y}_i^k}$. Thus, the total error incurred from first-order approximation, ignoring the higher-order terms, would be $\left((N_c - 1) \times (2\epsilon) \left| \frac{\partial \eta_i^{k(\text{int})}(x,y)}{\partial x} \right|_{x=\hat{x}_i^k, y=\hat{y}_i^k} \right)$, which establishes the desired result. ∎

Theorem 5.10 gives the conditions under which the knowledge about the intrusive forensics can be extended to semi non-intrusive forensics. The theorem also suggests that $\eta_i^{k(\text{int})} \geq \eta_i^{k(\text{semi})}$, and therefore the confidence score for parameter identification from semi non-intrusive forensics is lower than (or at most equal to) the ones that can be attained from intrusive forensics. This result is expected because intrusive forensic methodology gives more control than semi non-intrusive forensics, as the forensic analyst can break the device or system open to examine each of its individual components in greater detail. On the other hand, in the case of semi non-intrusive forensic analysis, the analyst would need to come up with good inputs to be given to the overall system and study the interactions between various system components based on the overall input/output response. Next, we examine the conditions when semi non-intrusive forensics and intrusive forensics provide the same accuracies.

Corollary 5.3 *The confidence scores for component parameter estimation via semi non-intrusive forensics is equal to the confidence scores for intrusive forensics when the knowledge of the component parameters do not help in the guessing the input x given the output y.*

Proof: From (5.66), we notice that equality among the confidence scores for semi non-intrusive forensics and intrusive forensics is obtained only when all the components in the system are $0-$consistent. A component is said to be $0-$consistent, by definition, when its input can be uniquely determined given its output, and viceversa. Further, for a $0-$consistent component, the knowledge of the component parameters do not help in the guessing the input x given the output y. This completes the proof. ∎

Corollary 5.4 *Intrusive analysis, semi non-intrusive forensic analysis, and completely non-intrusive forensic analysis provide the same confidence scores in parameter classification only when all the components in the device are* $0-$*consistent.*

Proof: The proof of this corollary follows from the proofs of Theorem 5.8 and Corollary 5.3. ∎

Comparing Corollary 5.2 and Corollary 5.4, we observe that the concept of *forensic independence* is equivalent to $0-$*consistency*. Further, it can be shown that if all the components of the device are forensically independent of each other, then all the components of the device are also $0-$consistent, and viceversa. This indicates the parallels between estimation and pattern classification theories.

Next, we re-consider the example discussed in Section 5.1.3 to illustrate pattern classification framework to forensically classify component parameters.

Example: Consider a system with the input-output response given by

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{5.67}$$

where $\underline{x} = [x_1 \ x_2]^T$ is the input to the system, $\underline{y} = [y_1 \ y_2]^T$ denotes the output from the system, $\underline{n} = [n_1 \ n_2]^T$ represents additive Gaussian noise with $E(n_1^2) = E(n_2^2) = \Sigma_n$ and $E(n_1 n_2) = 0$, and $\phi = [a_{11} \ a_{12} \ a_{21} \ a_{22}]^T$ are the component parameters. The goal of the forensic analyst is to compute the values of the parameter ϕ.

Contrary to the example in Section 5.1.3 where we assume that ϕ can take infinite possible values in the parameter space, in this example, we restrict the ϕ to take one of the two values in the parameter space Φ, *i.e.,* $\phi \in \Phi = \{\phi_1, \phi_2\}$. The parameter sets ϕ_1 and ϕ_2 are assumed to be of the form $\phi_1 = [\alpha_1 \ 0 \ 0 \ \alpha_2]^T$ and $\phi_2 = [0 \ \beta_1 \ \beta_2 \ 0]^T$, where the values of the parameters $\alpha_1, \alpha_2, \beta_1$, and β_2 are known

apriori. For our analysis, we assume that there is no apriori knowledge about the likelihood of choosing either ϕ_1 or ϕ_2 so that $p(\phi_1) = p(\phi_2) = 0.5$; and let ϕ_1 be the actual parameter set employed in the component without any loss in generality.

• **s-classifiability:** We first show that the component is semi non-intrusively classifiable. A component is *s-classifiable*, by definition, if for most inputs $\underline{x}$ and corresponding outputs $\underline{y}$,

$$p(\phi_1|\underline{y}, \underline{x}) \geq p(\phi_2|\underline{y}, \underline{x}). \tag{5.68}$$

Imposing the assumption that noise follows a Gaussian distribution, the requirement for s-classifiability in inequality (5.68) reduces to

$$(2\beta_1 x_2 - 2\alpha_1 x_1)y_1 + (2\beta_2 x_1 - 2\alpha_2 x_2)y_2 \leq (\beta_1^2 - \alpha_1^2)x_1^2 + (\beta_2^2 - \alpha_2^2)x_2^2. \tag{5.69}$$

This inequality indicates that the component is s-classifiable under the input $\underline{x} = [x_1\ x_2]^T$ with the actual parameter ϕ_1 if the output $\underline{y} = [y_1\ y_2]^T$ lies on the correct side of the straight line given by (5.69). Considering a specific case, if $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1$, the inequality in (5.69) reduces to $(y_2 - y_1)(x_2 - x_1) \geq 0$; suggesting that for an input $x_2 > x_1$, the component is s-classifiable under the hypothesis $\phi = \phi_1$, if the corresponding output satisfies $y_2 > y_1$. Now, we quantify the probability of $y_2 > y_1$ under the hypothesis $\phi = \phi_1$. It can be shown that

$$\begin{aligned} \Pr(y_2 > y_1|\phi_1) &= \Pr(n_2 - n_1 > x_1 - x_2|\phi_1) \\ &= \frac{1}{2}\left(1 + \text{erf}\left(\frac{x_2 - x_1}{2\Sigma_n}\right)\right). \end{aligned} \tag{5.70}$$

where 'erf' is the error function. When $x_2 > x_1$, the 'erf' term is approximately equal to '1' giving $\Pr(y_2 > y_1|\phi_1, x_2 > x_1) \approx 1$. Thus, for most inputs satisfying $x_2 > x_1$, the probability of deciding $\phi = \phi_1$ is close to '1'. Similarly, we can show that for inputs satisfying $x_2 < x_1$, $\Pr(y_2 < y_1|\phi_1, x_2 < x_1) \approx 1$; thus, establishing the s-classifiablility of the component for all range of inputs.

• **Confidence score and optimal inputs for semi non-intrusive forensics:** The confidence score attained via semi non-intrusive forensic analysis is given by (5.48) and can be reduced to

$$\begin{aligned}\eta^{(\text{semi})}(\underline{x},\underline{y}) &= 2p(\phi_1|\underline{y},\underline{x})-1,\\ &= \frac{p(\phi_1)p(\underline{y}|\phi_1,\underline{x})-p(\phi_2)p(\underline{y}|\phi_2,\underline{x})}{p(\phi_1)p(\underline{y}|\phi_1,\underline{x})+p(\phi_2)p(\underline{y}|\phi_2,\underline{x})}.\end{aligned} \tag{5.71}$$

As can be seen from the equation, the confidence score is a function of the input and can be improved by appropriate choice of the input. Under the condition that $p(\phi_1)=p(\phi_2)=0.5$, we get

$$\begin{aligned}\eta^{(\text{semi})}(x_1,x_2,y_1,y_2) &= \frac{\exp\left(-\frac{(y_1-\alpha_1x_1)^2+(y_2-\alpha_2x_2)^2}{2\Sigma_n}\right)-\exp\left(-\frac{(y_1-\beta_1x_2)^2+(y_2-\beta_2x_1)^2}{2\Sigma_n}\right)}{\exp\left(-\frac{(y_1-\alpha_1x_1)^2+(y_2-\alpha_2x_2)^2}{2\Sigma_n}\right)-\exp\left(-\frac{(y_1-\beta_1x_2)^2+(y_2-\beta_2x_1)^2}{2\Sigma_n}\right)}.\\ &= \frac{1-\exp(\mathcal{A}(x_1,x_2,y_1,y_2))}{1+\exp(\mathcal{A}(x_1,x_2,y_1,y_2))}.\end{aligned}$$

where

$$\mathcal{A}(x_1,x_2,y_1,y_2)=\frac{(y_1-\alpha_1x_1)^2+(y_2-\alpha_2x_2)^2-(y_1-\beta_1x_2)^2-(y_2-\beta_2x_1)^2}{2\Sigma_n}. \tag{5.72}$$

Optimal inputs can be computed by maximizing $\mathcal{A}(x_1,x_2,y_1,y_2)$ with respect to x_1 and x_2. For the specific case of $\alpha_1=\alpha_2=\beta_1=\beta_2=1$, the equation reduces to

$$\mathcal{A}(x_1,x_2,y_1,y_2)=\frac{(x_2-x_1)(y_2-y_1)}{2\Sigma_n}. \tag{5.73}$$

Therefore, the best input for semi non-intrusive forensics of this component is the one that maximizes $|x_2-x_1|$, *i.e.*, choose $x_1=\min_{x\in\Re_x}x$ and $x_2=\max_{x\in\Re_x}x$ or viceversa.

• **n-classifiability:** In this part, we assume that the input $\underline{x}$ follows a Gaussian distribution with mean $\mu_x=[\mu_1\ \mu_2]^T$ and $E(x_1^2)=E(x_2^2)=\Sigma_x$ with $E(x_1x_2)=0$.

Under this scenario, we have

$$\frac{p(\phi_1|\underline{y})}{p(\phi_2|\underline{y})} = \exp(\mathcal{B}(\underline{y})). \tag{5.74}$$

where

$$\mathcal{B}(\underline{y}) = \mathcal{B}(y_1, y_2) = \frac{(y_1 - \beta_1\mu_1)^2}{(\beta_1^2\Sigma_x + \Sigma_n)} + \frac{(y_2 - \beta_2\mu_2)^2}{(\beta_2^2\Sigma_x + \Sigma_n)} - \frac{(y_1 - \alpha_1\mu_1)^2}{(\alpha_1^2\Sigma_x + \Sigma_n)} - \frac{(y_2 - \alpha_2\mu_2)^2}{(\alpha_2^2\Sigma_x + \Sigma_n)} \tag{5.75}$$

If $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1$, then $\mathcal{B}(\underline{y}) = 0$ and $p(\phi_1|\underline{y}) = p(\phi_2|\underline{y})$. The component is not non-intrusively classifiable under this scenario as there exists no input $\underline{x}^*$ for which the corresponding output $\underline{y}^*$ satisfies $p(\phi_1|\underline{y}^*) > p(\phi_2|\underline{y}^*)$. ∎

5.2.3 A Note on the Definition of Confidence Score

We define the confidence score for parameter estimation according to (5.48) as the difference between the probability of correct classification and the average of the corresponding likelihoods of the making a wrong decision as

$$\eta_i^{(1)}(x, y) = p(\phi_i|y, x) - \frac{1}{N_a - 1} \sum_{j=1,2,\ldots,N_a, j\neq i} p(\phi_j|y, x). \tag{5.76}$$

Several other definitions for the confidence score in classification have been proposed in literature and have been employed in practice to judge the confidence in classification. In [139], Wan defined confidence score as the Kullback-Leibler distance between the estimated probability density function and uniform distribution as

$$\eta_i^{(2)}(x, y) = \mathrm{D}(p(\phi_i|y, x)||\mathcal{U}), \tag{5.77}$$

where $\mathcal{U}$ represents uniform distribution. The equation can therefore be reduced to

$$\eta_i^{(2)}(x, y) = \sum_{i=1}^{N_a} p(\phi_i|y, x) \log(N_a p(\phi_i|y, x)). \tag{5.78}$$

In the Chapter 3, we developed a confidence score based on the symmetric Kullback-Leibler divergence as [128]

$$\begin{aligned}\eta_i^{(3)}(x,y) &= \mathrm{D}(p(\phi_i|y,x)||\mathcal{U}) + \mathrm{D}(\mathcal{U}||p(\phi_i|y,x)),\\ &= \sum_{i=1}^{N_a}\left(p(\phi_i|y,x) - \frac{1}{N_a}\right)\log(N_a p(\phi_i|y,x)),\end{aligned} \tag{5.79}$$

and in [130], we defined a confidence metric as the difference between the probability of correct classification and the maximum of the corresponding likelihoods of the making a wrong decision:

$$\eta_i^{(4)}(x,y) = p(\phi_i|y,x) - \max_{j=1,2,\ldots,N_a, j\neq i} p(\phi_j|y,x). \tag{5.80}$$

Although the definitions of confidence score in $\eta_i^{(1)}(x,y)$ to $\eta_i^{(4)}(x,y)$ are different, they provide different approaches to evaluate the goodness in decision making and can provide different insights into the classification result. However, many of the theorems, corollaries, and lemmas derived and proved proved in Section 5.2.2 are fundamental and hold true invariant of the choice of the confidence score metric as shown in the following example.

Example: In this example, we show that the result in Theorem 5.7 holds true even a different choice of confidence measure. Specifically, we consider the case $\eta_i(x,y) = \eta_i^{(2)}(x,y)$. The confidence score for semi non-intrusive forensics and completely non-intrusive forensics for this case are given by

$$\eta_i^{\mathrm{semi}}(x,y) = \log(N_a) + \sum_{i=1}^{N_a} p(\phi_i|y,x)\log(p(\phi_i|y,x)). \tag{5.81}$$

$$\eta_i^{\mathrm{non}}(y) = \log(N_a) + \sum_{i=1}^{N_a} p(\phi_i|y)\log(p(\phi_i|y)). \tag{5.82}$$

To show that $\eta_i^{\mathrm{semi}}(x,y) \geq \eta_i^{\mathrm{non}}(y)$, we start with the identity: $E(p(\phi_i|y,x)) = p(\phi_i|y)$. This identity implies that there exists at least one input $x_0 \in \Re_x$ for

which $p(\phi_i|y, x_0) \geq p(\phi_i|y)$, and therefore for this input $\eta_i^{\text{non}}(y) \leq \eta_i^{\text{semi}}(x_0, y) \leq \eta_i^{\text{semi}}(\hat{x}, \hat{y}) = \eta_i^{\text{semi}}$. Here, $\hat{x}$ is the optimal input to the component with $\hat{y}$ denoting its corresponding output. ∎

5.3 Chapter Summary

In this chapter, we develop two new theoretical frameworks for analyzing information forensics to analyze component forensics depending on the nature of the component. In the first scenario, we assume that the parameter values of a component can take infinite number of possibilities. Under this scenario, we introduce a framework based on estimation theory, Fisher information, and the Cramer-Rao lower bound. We define formal notions of identifiability of components under intrusive, semi non-intrusive, and completely non-intrusive forensic analysis cases and quantify the accuracies at which the component parameters can be estimated in each case using Fisher information as a criterion.

In the second scenario, we assume that the forensic analyst has some apriori knowledge about the component and has information about the possible superset of parameter values employed in the component. For this scenario, we employ ideas from pattern classification theory to answer forensic questions about what components and processing operations are classifiable and what are not; and quantify the confidence in which the component parameters can be classified.

Building on the proposed theoretical analysis frameworks, we establish a number of fundamental results. Our theoretical analysis suggests that intrusive forensics gives superior estimation accuracies and classification confidence over semi non-intrusive forensics, and this is better than completely non-intrusive scenario. We demonstrate that the accuracy in estimating the component parameter and

the confidence in classifying the component algorithms depend on the nature of available inputs and testing conditions, and can be improved by better choice of inputs. We then apply the theoretical framework in case studies to design optimal inputs for semi non-intrusive forensics; and show that the confidence in parameter identification can be improved via such an approach. The proposed theoretical model can also be extended to study post-device processing operations such as tampering, and to provide a theoretical foundation for media forensics to answer a number of forensic questions related to who has done what to the content, when, and how.

Chapter 6

Case Studies and Applications of Theoretical Forensics Framework

In this chapter, we present case studies and applications of the proposed theoretical analysis frameworks presented in Chapter 5. Specifically, we focus on the problem of semi non-intrusive forensics. We briefly describe the imaging model in digital cameras and define the notations used in Section 6.1. In Section 6.2, we show that the parameters of such important components as color interpolation and white balancing can be better estimated via semi non-intrusive forensics compared to completely non-intrusive forensics. Based on a detailed modeling of the imaging process and knowledge of the possible algorithms employed in such components as color interpolation and white balancing, in Section 6.3, we design a heuristic input for semi non-intrusive forensics of digital camera components and show that the designed pattern can provide better accuracies. The pattern is then optimized in Section 6.4 using metrics from theoretical analysis and simulation results are presented to demonstrate the goodness of the pattern. The chapter is summarized in Section 6.5.

To our best knowledge, this is the first work to address the problem of semi non-intrusive component forensics. Related work fall into two basic categories. In the forensics literature, there have been work that aim to find the parameters of post-camera processing operations [84,110] such as JPEG compression, resampling, and brightness change; and to non-intrusively estimate the parameters of camera components such as lens distortions, color filter array [123], and color interpolation [112,123]. However, the accuracy of these non-intrusive techniques is limited by the nature of the available data. A second group of prior art concerns television and camera manufacturing technologies. Among these work, there have been studies that focus on designing test patterns to tune the parameter settings of television sets by analyzing its response to specific inputs [89]. However, these work are not intended for estimating the parameters of internal device components.

6.1 Signal Processing Model of Camera Components

In this section, we develop a signal processing model of camera components. Figure 2.1 shows the image acquisition model in digital cameras. Let x be the input to the camera's color interpolation module. For our work, we divide the image into different types of regions based on the local gradient directions, and approximate color interpolation in each region to be linear.[1] The output y_1 after color interpolation can be written as

$$y_1(m,n,c) = \sum_{k,l} \alpha(k,l,c)x(m-k,n-l,c) + n_1(m,n,c), \tag{6.1}$$

[1]In Chapter 3, we show that this linear approximation is good for estimating the color interpolation coefficients.

for each texture region. Here, α denotes the color interpolation coefficients and the summations over variables k and l are done in the regions where the filter $\alpha(k, l, c)$ has support. The noise term $n_1(m, n, c)$ is used to simulate the model fitting error, and in our analysis, we assume that n_1 follows a Gaussian distribution.

After color interpolation, the interpolated image y_1 undergoes white balancing to give y_2. White balancing and color correction are typically done in the camera as part of the post-processing block to remove unrealistic color casts from the image. White balancing is typically multiplicative in nature, where the output is obtained by scaling the input by the chosen scaling factor. In manual white balancing, the user chooses the appropriate multiplication constants for each color channel so that a white colored object looks white after compensation. On the other hand, auto white balancing algorithms compute the multiplication factors based on the estimated illuminance of the scene [7] and use these estimates for scaling the input. White balancing operations can be mathematically represented as

$$y_2(m, n, c) = \sum_{j=1}^{3} \beta(c, j) y_1(m, n, j), \text{ for } c = 1, 2, 3. \tag{6.2}$$

where β are the white balancing coefficients.

Finally, the image may be JPEG compressed to reduce storage space. Compression can be modeled as quantization in the DCT domain, and can be represented as additive noise in the pixel domain. Denoting this compression noise as n_2, the final image is given by

$$y(m, n, c) = y_2(m, n, i) + n_2(m, n, c). \tag{6.3}$$

Combining (6.1), (6.2), and (6.3), we obtain the input-output response of the

digital camera

$$y(m,n,c) = \sum_{j=1}^{3} \sum_{k,l} \alpha(k,l,c)\beta(i,j)x(m-k,n-l,c) + \sum_{j=1}^{3} \beta(i,j)n_1(m,n,c) + n_2(m,n,c). \tag{6.4}$$

The goal of the forensic analyst is now to estimate the device parameters $\alpha(.,.,.)$ and $\beta(.,.)$.

6.2 Theoretical Analysis of Digital Camera Components

In this section, we employ the theoretical frameworks presented in Chapter 5 to analyze the parameters of such camera components as color interpolation, white balancing, and JPEG compression. We analyze these components from both estimation and pattern classification perspectives and determine the accuracies in computing the component parameters.

6.2.1 Color Interpolation

Color interpolation is an important processing stage in digital cameras. Most cameras of different brands/models employ a different algorithm for color interpolation and therefore estimating the interpolation parameters provides very useful information to build a robust camera identifier as shown in Chapter 3 and [128]. In this subsection, we examine the conditions under which the color interpolation component parameters are identifiable.

Typically, the data recorded by the CFA are interpolated using its neighboring pixel values to form the interpolated image as represented by equation (6.1). Obtaining the component parameters α in a general case involves solving a blind

deconvolution problem. However, additional information about the sampling pattern could be used to simplify the problem as the knowledge of the CFA gives the locations of the set of pixels that are interpolated and those that are directly obtained from the CCD sensor. With this information and with the assumption that the color interpolation coefficients, α, has support in the range $[-\lfloor\frac{N_\alpha}{2}\rfloor, \lfloor\frac{N_\alpha}{2}\rfloor] \times [-\lfloor\frac{N_\alpha}{2}\rfloor, \lfloor\frac{N_\alpha}{2}\rfloor] \times [1, 3]$, (6.1) can be equivalently re-written for the 'red' color under no-noise case as

$$\begin{bmatrix} y_1(1,1,1) \\ y_1(1,2,1) \\ y_1(1,3,1) \\ y_1(1,4,1) \\ \vdots \\ y_1(W,H,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ \alpha(0,1,1) & 0 & \alpha(0,-1,1) & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ \alpha(0,3,1) & 0 & \alpha(0,1,1) & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \dots \end{bmatrix} \begin{bmatrix} x(1,1,1) \\ x(1,2,1) \\ x(1,3,1) \\ x(1,4,1) \\ \vdots \\ x(W,H,1) \end{bmatrix}, \quad (6.5)$$

where W and H denote the width and the height of the image. In constructing these equations, we assume that the camera employs Bayer CFA [13] to sample the real-world scene and similar equations can be obtained for other CFA.

In the absence of post-interpolation processing, such as white balancing and JPEG compression, there would be no additive noise and $y = y_1$. Further, under these conditions, the values of the camera output image at locations corresponding to $\{y_1(1,1,1), y_1(1,3,1), y_1(1,5,1), \dots\}$ are obtained directly from the 'red' color component of camera input, and the values at the remaining intermediate pixel locations corresponding to $\{y_1(1,2,1), y_(1,4,1), y_1(1,6,1), \dots\}$ are obtained interpolated. Therefore, with the knowledge of the color filter array, the output $y_1 = y$

gives complete information about the input and we obtain

$$\begin{bmatrix} y_1(1,2,1) \\ y_1(1,4,1) \\ \vdots \end{bmatrix} = \begin{bmatrix} \alpha(0,1,1) & \alpha(0,-1,1) & \dots \\ \alpha(0,3,1) & \alpha(0,1,1) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} y_1(1,1,1) \\ y_1(1,3,1) \\ \vdots \end{bmatrix}. \tag{6.6}$$

This final set of equations in (6.6) are dependent only on the camera outputs and can be solved by least squares method to estimate the component parameters. Therefore, in the absence of noise and post-interpolation processing, the average error in estimating the cameras' color interpolation parameters with an input x via semi non-intrusive forensics is equal to the average estimation error obtained via completely non-intrusive forensics with the knowledge of just the component output y_1, *i.e.*, $\Delta_s(x) = \Delta_n$.

Equation (6.6) also suggests that the component is n-classifiable, s-classifiable, and i-classifiable in the absence of noise and post-interpolation processing. Color interpolation component is therefore a particular example of a component for which n-classifiability implies s-classifiability which is not true in a general case. This property of color interpolation can be attributed to the fact that the component is $0-$consistent, and the knowledge of the output y gives full information about the input x, and $p(y|\phi_i) = p(y|\phi_i, x)$, where ϕ_i are the component parameters.

In the presence of noise and post-interpolation processing, the component would no longer be $0-$consistent and semi non-intrusive analysis would provide better accuracies than completely non-intrusive analysis. In the subsequent sections, we design a heuristic input and optimize it to increase the estimation accuracy and classification confidence in computing the color interpolation parameters.

6.2.2 White Balancing

In this part, we theoretically analyze the white balancing component under the presence and absence of additive noise. We begin with the 'no-noise' case.

• **No Noise case:** The input-output relationship for the white-balancing operation under no noise is given by (6.2) and can be expressed in the matrix form as

$$\mathbf{y} = \mathbf{y}_2 = \theta \mathbf{y}_1, \tag{6.7}$$

where θ is the white balancing parameter, and $\mathbf{y}_1$ and $\mathbf{y}$ represent the input and the output of the white balancing component, respectively.

In the noiseless case case, the component is i-classifiable and s-classifiable because the forensic analyst can accurately estimate θ given one instantiation of the input and output, as $\theta = \mathbf{y} \times \mathbf{y}_1^{-1}$. However, the component may not be n-classifiable in a general scenario because, in the absence of the knowledge about the input $\mathbf{y}_1$, the values of $\mathbf{y}_1$ and θ may be appropriately swapped and the information about the output $\mathbf{y}$ would not resolve the ambiguity.[2]

• **Under Additive Noise:** Most often, white balancing precedes processing such as JPEG compression in digital cameras. Operations such compression add noise to the final output and under this scenario, and therefore the final output can be written as

$$\mathbf{y} = \theta \mathbf{y}_1 + \mathbf{n}_2, \tag{6.8}$$

where $\mathbf{n}_2$ models the additive noise.

To simplify mathematical analysis of the white balancing component, we consider a specific case with $\mathbf{y}_1 = y_1$ of unit length and $E(n_2^2) = \sigma_n$. Under this

[2]The white balancing component may be n-classifiable if there is a restriction on the parameter space Θ and/or the input space $\Re_x$ that would help resolve the ambiguity.

scenario,

$$p(y|y_1,\theta) = \frac{1}{(2\pi\sigma_n^2)^{1/2}} \exp\left(-\frac{(y-\theta y_1)^2}{2\sigma_n^2}\right), \tag{6.9}$$

and the Fisher information for semi non-intrusive forensics can be derived as:

$$\mathcal{I}_s(y_1,\theta) = \frac{y_1^2}{\sigma_n^2}. \tag{6.10}$$

This equation suggests that the Fisher information is equal to the signal to noise ratio (SNR); this satisfies intuition as we notice that as the SNR increases, the Fisher information increases and the overall accuracy improves.

With the assumption that the input to the component, y_1, follows a Gaussian distribution with mean μ_{y_1} and variance $\sigma_{y_1}^2$, the pdf of the output y can be computed as

$$p(y|\theta) = \frac{1}{\sqrt{2\pi(\theta^2\sigma_{y_1}^2+\sigma_n^2)}} \exp\left(-\frac{(y-\theta\mu_{y_1})^2}{2(\theta^2\sigma_{y_1}^2+\sigma_n^2)}\right) \tag{6.11}$$

and the Fisher for completely non-intrusive forensics can be calculated to be

$$\mathcal{I}_n(\theta) = \frac{3\theta^2\sigma_{y_1}^4(\theta^2\sigma_{y_1}^2+\sigma_n^2)^2 + \mu_{y_1}^2(\theta^2\sigma_{y_1}^2+\sigma_n^2) + 2\theta^2\sigma_{y_1}^4}{(\theta^2\sigma_{y_1}^2+\sigma_n^2)^2}. \tag{6.12}$$

Comparing (6.10) and (6.12), we notice that for any input y_1 that satisfies $y_1 \geq \sigma_n\sqrt{\mathcal{I}_n(\theta)}$, the Fisher information for semi non-intrusive forensics would be higher than the Fisher information for completely non-intrusive forensics. Therefore, by choosing such an input, the overall estimation errors obtained via semi non-intrusive forensics can be made lower compared to non-intrusive studies. Thus, the white balancing parameters can be better estimated semi non-intrusively by appropriate choice of inputs.

6.2.3 JPEG compression

JPEG compression can be considered as quantization in the Discrete Cosine Transform (DCT) domain. The compression parameters and the quality factors can be

reasonably estimated via statistical analysis based on binning techniques just based on the output image [33,84]. Therefore, the component is n-classifiable for non-zero inputs.

6.3 Semi Non-Intrusive Forensics with Heuristic Pattern

In the previous section, we have shown that the parameters of such important components as color interpolation and white balancing can be estimated with a higher accuracy and confidence via semi non-intrusive forensics compared to completely non-intrusive forensics. In this section, we design a heuristic pattern for semi non-intrusive forensics of digital cameras and show that the heuristic pattern can provide better accuracies in parameter estimation.

6.3.1 Heuristic Pattern Design

Lets consider the imaging model discussed in Section 6.1. Concatenating all the elements of $y(m, n, c)$ to form $\mathbf{y}$, and representing (6.4) in matrix form, we obtain

$$\mathbf{y} = A_{\alpha\beta}\mathbf{x} + B_{\beta}\mathbf{n}_1 + \mathbf{n}_2. \tag{6.13}$$

where $A_{\alpha\beta}$ and B_{β} denote the matrices of appropriate dimension and are formed from the parameters α and β. The sub-scripts in these matrices are used to indicate their dependence on the appropriate component parameters. The goal of the forensic analyst in semi non-intrusive forensics is to design an input that would help increase the confidence (or accuracy) in classifying (or estimating) the device parameters $\phi = [A_{\alpha\beta}\ B_{\beta}]$.

Suppose N_a is the total number of possible algorithms employed by the component such that $\phi \in \Phi = \{\phi_1, \phi_2, \ldots, \phi_{N_a}\}$, the forensic analyst computes the optimal input as the one that maximizes the confidence score

$$\eta_i(x) = \frac{N_a}{N_a - 1}\left(p(\phi_i|x, y) - \frac{1}{N_a}\right). \tag{6.14}$$

Assuming the noise terms n_1 and n_2 to be independent and Gaussian distributed with mean zero and variance $\sigma_{n_1}^2$ and $\sigma_{n_2}^2$, respectively, it can be shown that finding an input that maximizes (6.14) is equivalent to computing the input that maximizes the distance, $(A_{\alpha\beta}(i) - A_{\alpha\beta}(j))\mathbf{x}$, between the means of every two pairs of distributions. In this subsection, we develop heuristics to achieve this property.

As seen from the analysis, choosing the *optimal* input pattern would depend on the nature of the algorithms in the parameter space Φ. In the case of the color interpolation component, the algorithm space Φ can be mainly classified into two categories as adaptive and non-adaptive methods depending on the way they handle edge regions (see Appendix I of Chapter 3 for a brief summary). Therefore, a good input to identify the interpolation category would be a pattern with significant edge patterns, either in the horizontal or vertical direction. A sample is shown in Figure 6.1(a). The corresponding images interpolated with non-adaptive and adaptive methods are shown in Figure 6.1(b) and (c) and their magnified versions are shown in Figure 6.1(d) and (e) respectively. As can be seen from the figures, there are significant artifacts for images interpolated using non-adaptive methods, and no such distortions are present in the images interpolated using gradient based adaptive techniques. This result is expected because the non-adaptive methods to not use any kind of edge sensing algorithms to avoid averaging across the edge. In this case, we would be able to easily distinguish between the two kinds of interpolation methods only by visually examining the outputs under

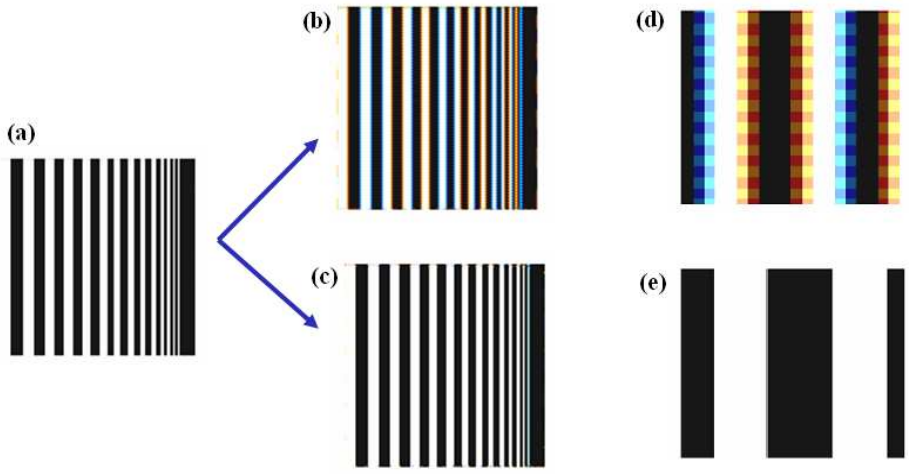


Figure 6.1: A possible input pattern to identify the interpolation type. The figure shows (a) sample input pattern; (b) image obtained after non-adaptive interpolation techniques; (c) image obtained after edge based adaptive methods; (d) a magnified version of (b) showing the artifacts; (e) a magnified version of image in (c).

this input. This illustration indicates that the choice of an optimal input would in general depend on the type of possible interpolation algorithms that we intend to identify (or differentiate). For instance, the sample pattern in Figure 6.1 may not be able to distinguish between two different types of adaptive methods that use different set of coefficients for interpolation.

Generalizing on this observation, we define a set of properties required for an optimal input pattern based on a detailed study of the imaging process and possible algorithms employed in each component.

- *Identifying Color Interpolation Methods:*
  - To help distinguish between different kinds of adaptive interpolation methods, it would be necessary to study the similarity and differences

in the way each of the interpolation methods handle different types of directional edges. Thus, a converging wedge pattern as shown in Figure 6.2 would be useful.

– Chirp signals can be used to capture the variations in the frequency domain as they have been known to have a very good frequency response. The basic equation for generating a chirp signal is of the form

$$s(m,n) = a_1 \cos(a_2 m^2 + a_3 n^2).$$

where a_1, a_2, and a_3 are suitably chosen constants. These patterns also provide us with a simple method to construct symmetric and circular patterns with gradually decreasing widths and thickness, and in turn facilitating performance studies of the interpolation methods under various frequency levels.

– Some interpolation methods have different ways to handle smooth regions. Generally, bilinear or bicubic interpolation methods are used in smooth regions due to their ease in implementation and because they do not produce pronounced visual distortions in these areas. Thus, the ideal pattern should also have reasonable sized smooth and gradually varying regions to help identify the type of interpolation used here.

– *Naturalness:* Many of the interpolation methods are designed to work well for natural images taken using a camera with a gradually changing smooth hue. Some of them further assume that the edges of the three color channels are aligned, and some others suppose that the differences between the color channels (red-green, red-blue, blue-green) are continuous. Hence, it would be necessary that have a smooth hue in order to achieve maximum accuracy in identification.

- *Identifying White Balancing methods:* Most of the cameras use white-patch algorithm or the grey-world methods for white balancing. The white patch method is based on relative normalization of the individual color channels based on assumption that a particular region (in the image) is white. Thus, introducing large sections of all-black and all-white regions with constant intensity would enable us to find if the white-patch methods were used. To identify the grey-world algorithms, it would be necessary to see if the average pixel value in the output image is close to the mid-grey value of 128.

- *Identifying Gamma Correction:* The best input pattern to find the value of Gamma is the varying grey scale pattern. Thus, comparing the output grey scale values with the input, one can obtain a very good and reasonable estimate of the value of the parameter gamma.

- *Identifying Lens distortions:* The best pattern to help identify any kind of lens distortions is the checkerboard pattern with long straight lines. We would also be able to estimate the parameters of the lens distortions by studying the transformations undergone by a straight line. The checkerboard pattern also helps align the captured image with the original image.

Based on the requirements outlined above, a possible input pattern is constructed as shown in Figure 6.3 by combining different patterns each satisfying some of the requirements listed above. As can be seen from the figure, it has the variable frequency chirp patterns at the center, the wedge patterns have been repeated twice to help provide more information about the variability in handling gradients along different directions. Gradually changing smooth regions border the chirp patterns to help identify the interpolation methods used in smooth regions. The image has been post-processed by fine tuning the hue and the ratios red/green

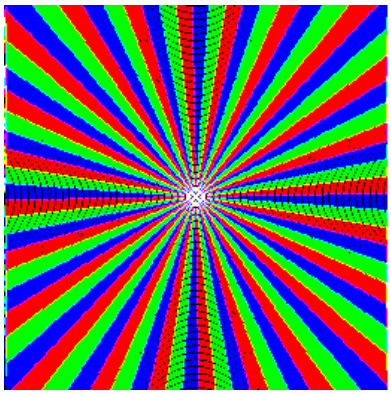


Figure 6.2: Wedge patterns for semi non-intrusive forensics.

and the blue/green components have been smoothened to introduce naturalness. Finally, the difference images (red-green and blue-green) have also been spatially averaged to obtain good performance.

6.3.2 Component Forensics Analysis of Color Interpolation

As shown in Section 6.2.1, in the absence of noise and post-interpolation processing operations, color interpolation module is 0−consistent and completely non-intrusive analysis would provide the same accuracies as semi non-intrusive analysis and the knowledge of the input does not provide any additional information to aid forensic analysis in this case. However, in the presence of noise, the component would no longer be 0−consistent and semi non-intrusive analysis would provide better accuracies than completely non-intrusive analysis. In this subsection, we examine the effectiveness of the heuristic input pattern for semi non-intrusive forensics of color interpolation module and compare the results obtained with natural images under completely non-intrusive forensics scenario in the presence of post-interpolation processing.

We employ the proposed heuristic pattern for semi non-intrusive forensic anal-

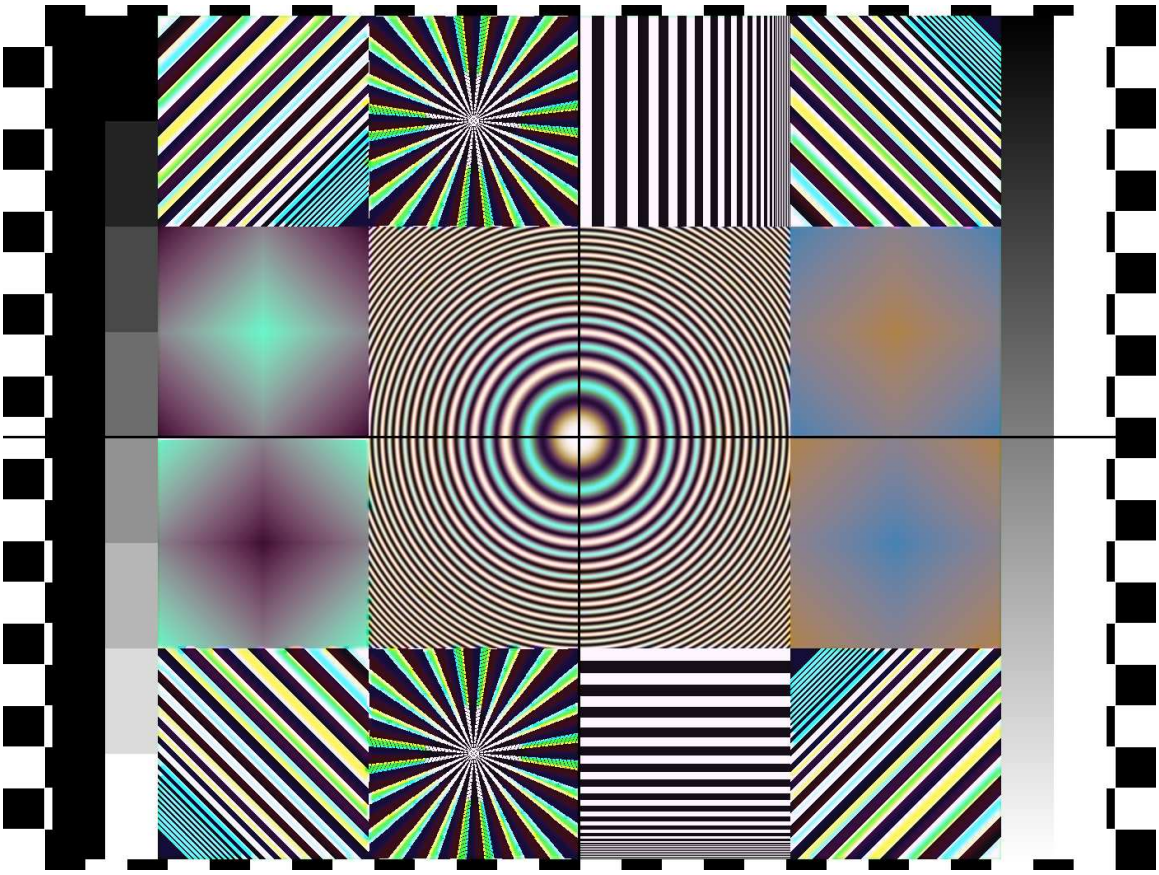


Figure 6.3: Heuristically designed input pattern.

ysis. In order to simulate completely non-intrusive forensic scenario for comparison studies, we select 20 representative images corresponding to different natural scenes [129, 133]. These images are first down-sampled to remove the effects of previously applied filtering and interpolation operations, sampled on the Bayer filter array [13], and then interpolated using six different interpolation algorithms to reproduce the scene capture process in cameras. The interpolation methods that we consider are: (a) Linear types of interpolation, including Bilinear and Bicubic, and (b) Non-linear interpolation methods including Smooth Hue, Median Filter based approach, Gradient based, and Adaptive Color Plane [7]. These 120 images obtained using these six different interpolation techniques form the non-intrusive

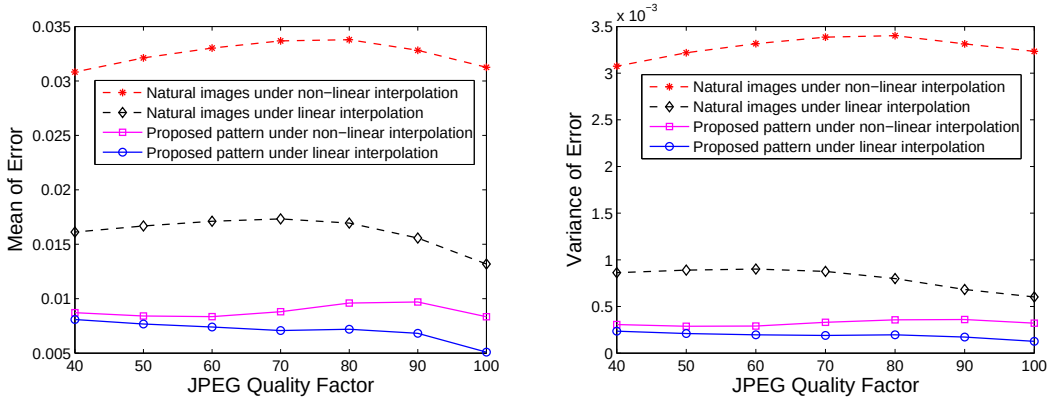


Figure 6.4: Results for color interpolation showing (a) mean and (b) variance of estimation error.

forensic dataset. We test the efficiency of semi non-intrusive forensics from both an estimation and pattern classification perspective.

Performance Evaluation from Estimation Perspective

For each image in the dataset, we estimate the interpolation coefficients from each type of region $\Re_m (m = 1, 2, 3)$ by solving the least squares problem [128], re-interpolate the image using the estimated coefficients, and find the estimation error. We compare the estimation results obtained semi non-intrusively using the proposed heuristic pattern with the ones got by employing natural images under non-intrusive scenarios. Figure 6.4(a) and (b) compare the results in terms of the mean and variance of the estimation error, respectively, for the two linear and four non-linear interpolation algorithms. As can be seen in the figure, the proposed heuristic pattern gives an average estimation error close to 0.007 per pixel that is much lower compared to natural images for which the values are around $0.015 - 0.03$. This suggests the effectiveness of the proposed heuristic pattern for improving the estimation of the color interpolation coefficients and demonstrates

the performance gains of semi non-intrusive forensics over the completely non-intrusive scenario.

Performance Evaluation from Classification Perspective

In this part, we study the performance of the heuristic pattern for classifying the interpolation type. For our experiments, we estimate the interpolation coefficients from each of the 120 synthetic images in the dataset and classify them with a SVM classifier [148]. We compute the confidence value as a difference between the probability of correct classification and the maximum of the corresponding likelihoods of the making a wrong decision, *i.e.,*

$$\eta_i(x,y) = p(\phi_i|y,x) - \max_{j=1,2,\ldots,N_a, j\neq i} p(\phi_j|y,x), \tag{6.15}$$

and use this as a metric to examine the classification results.

We study the robustness in parameter classification under JPEG compression. In Table 6.1, we show the confidence scores obtained on 'correct' classification under different quality levels of JPEG compression. We note that the maximum confidence is attained under 'no compression' for most of interpolation algorithms, and the confidence score reduces as the JPEG quality factor reduces. The '$*$' marks in the table under low JPEG quality indicate mis-classification. Upon a closer look at these results, we find that these bilinear and smooth hue interpolated images have been wrongly classified as bicubic. This result is expected because bilinear and bicubic employ very similar interpolation approaches, and smooth hue uses bicubic for the 'green' component as discussed in Appendix I of Chapter 3. The confidence values obtained for the heuristic pattern, in all scenarios, are significantly higher than those obtained for natural images which are in the range of $50-60\%$ even under 100% JPEG quality. This demonstrates the superiority of

Table 6.1: Variation of the classification confidence score as a function of JPEG quality factor for the heuristic pattern in Figure 6.3. $*$ indicates mis-classification.

Algorithm	No Compr.	90%	80%	70%	60%	50%	40%	30%
Bilinear	74%	68%	35%	*	*	*	*	*
Bicubic	77%	39%	55%	65%	60%	44%	25%	3%
Smooth Hue	94%	25%	16%	*	*	*	*	*
Median Based	64%	72%	68%	73%	77%	78%	67%	29%
Gradient Based	99%	92%	89%	83%	76%	71%	66%	69%
ACP	87%	50%	35%	22%	27%	25%	14%	*

the designed pattern for semi non-intrusive analysis.

A Closer Look at Estimation and Classification Results

We take a closer look at the estimation and the classification results to understand the reasons for superior performance. More specifically, we divide the heuristic pattern, shown in Figure 6.3 into various 512×512 regions depending on the location of wedge, chirp, horizontal, and vertical gradient patterns. The image blocks are then interpolated with each of the 6 different interpolation methods, and the interpolation coefficients are estimated from these blocks for classification. In Figure 6.5 (a)–(f), we show the images obtained from the six interpolation algorithms and highlight in green the regions that have been correctly classified by the SVM classifier. For instance, when interpolated with the bilinear method, all the regions except the wedge regions and the horizontal/vertical gradient regions are correctly classified to be bilinearly interpolated and the remaining regions were mis-classified.

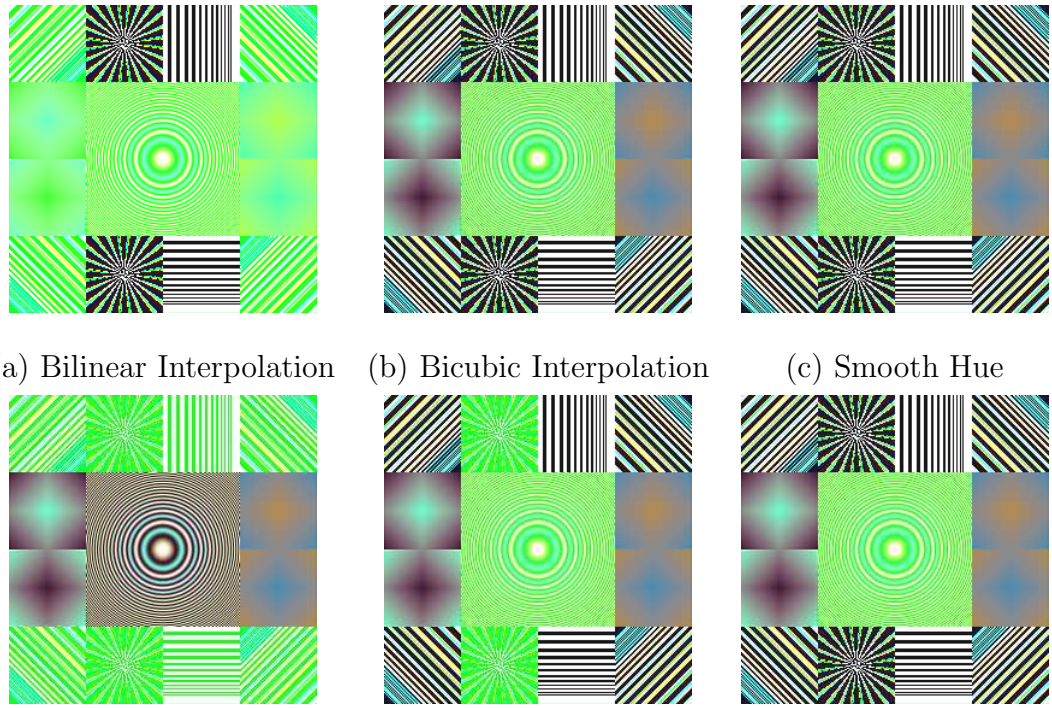
Comparing the highlighted regions in all the six images, we note that different types of regions are correctly classified when interpolated with different techniques. For example, the chirp patterns in the center can help identify the bilinear, bicubic, smooth hue, gradient based, and adaptive color plane methods. However, they may not be very good for identifying median based methods. On the other hand, converging wedge patterns are very good in identifying the median interpolation and gradient based methods. The horizontal and vertical gradient patterns can help distinguish adaptive versus non-adaptive methods, but cannot help separate two different types of adaptive methods or two different kinds of non-adaptive methods. Thus, our results indicate that while the individual patterns may not be separately good for identifying the exact interpolation algorithm, the proposed heuristic pattern is very good. When the entire image is given as an input, the coefficients obtained from each of the regions contribute to improve the overall classification accuracy; thus, improving the confidence in forensic analysis.

6.3.3 Forensics Analysis of White Balancing Parameters

In this subsection, we focus on white balance parameter estimation. We begin by describing the estimation algorithm and then present simulation results and analysis.

Proposed Algorithm to Estimate White Balance Parameters

A brief survey of white balancing methods are included in Appendix of this chapter. White balancing operations are typically multiplicative [39, 150] as shown in (6.2) and each color in the photograph is multiplied by an appropriately chosen constant in the camera color space. Using U to represent the transformation matrix that



(a) Bilinear Interpolation (b) Bicubic Interpolation (c) Smooth Hue

(d) Median based (e) Gradient based (f) ACP

Figure 6.5: A closer look at the heuristic pattern highlighting the regions that are correctly classified under different types of color interpolation algorithms.

is used to convert the RGB color coefficients to camera color space, the white balancing operation can be modeled as

$$\begin{bmatrix} y_2(m,n,1) \\ y_2(m,n,2) \\ y_2(m,n,3) \end{bmatrix} = U^{-1}\Lambda U \begin{bmatrix} y_1(m,n,1) \\ y_1(m,n,2) \\ y_1(m,n,3) \end{bmatrix}, \tag{6.16}$$

where $y_1(.,.,.)$ represents the raw pixels, $y_2(.,.,.)$ represents the white-balanced pixels, and the 3×3 diagonal matrix Λ denotes the white-balancing coefficients that are chosen based on the lighting conditions of the scene.[3] In most commercial

[3]Diagonal transformation matrix is preferred for Λ as it follows the Von-Kries hypothesis [39], and has only 3 parameters to be estimated from the scene.

cameras, white balancing is done in the XYZ color space [150], and U in this case would correspond to the color transformation from RGB to XYZ space. Some modern digital cameras may perform sensor sharpening, and appropriate modifications are done to the matrix U to include these effects. Some sample values of the transformation matrix, U, for FujiFilm FinePix S5000 and Canon EOS Digital Rebel are shown in Figure 6.6(a) and (b), respectively. Note that U is tied to a camera, while the value of Λ varies for each picture taken by the device.

As shown in Section 6.2, it would be difficult to non-intrusively estimate the white balancing parameters U and Λ accurately from the output images without the knowledge of the actual raw values captured by the sensor. However, they can be semi non-intrusively estimated. If the digital camera can produce raw images, the pixel values as captured by the CCD sensors can be read out from the captured image. These values can be used alongwith the actual white balanced output to estimate U and Λ by solving (6.16). For digital cameras that do not produce the raw format, the values of U can be estimated by a two-step process [129,133]. The first step obtains two images with approximately the same raw data but different white balanced processed versions. This can be done by manually choosing different built-in white balancing options while taking the pictures, for example, one image with white balancing setting fixed to "tube light" and another with "tungsten light." Let the white balanced RGB pixel values in the first image be denoted as $R_{wb}^{(1)}$, $G_{wb}^{(1)}$, and $B_{wb}^{(1)}$ and let $R_{wb}^{(2)}$, $G_{wb}^{(2)}$, and $B_{wb}^{(2)}$ represent the corresponding values in the second image. Denoting the corresponding white balancing constants employed in generating the two images by $\Lambda^{(1)}$ and $\Lambda^{(2)}$, respectively, we can show

1.503428	-0.424598	-0.078830	1.591484	-0.645577	0.054094
-0.056807	1.369831	-0.313025	-0.083807	1.479398	-0.395591
0.032900	-0.403764	1.370864	0.069723	-0.473899	1.404176
(a) FujiFilm FinePix S5000			(b) Canon EOS Digital Rebel		

Figure 6.6: Actual values of the transformation matrix (U) for two different camera brands.

that

$$\begin{bmatrix} R_{wb}^{(2)} \\ G_{wb}^{(2)} \\ B_{wb}^{(2)} \end{bmatrix} = U^{-1}(\Lambda^{(2)}/\Lambda^{(1)})U \begin{bmatrix} R_{wb}^{(1)} \\ G_{wb}^{(1)} \\ B_{wb}^{(1)} \end{bmatrix}. \tag{6.17}$$

Here, the notation $\Lambda^{(2)}/\Lambda^{(1)}$ represents a diagonal matrix with each diagonal element obtained as an element-wise division of the corresponding terms in $\Lambda^{(2)}$ and $\Lambda^{(1)}$.

In the following, we test our proposed estimation techniques for simulated data and study its robustness to JPEG compression with both synthetic data and actual images taken from the camera.

Testing with Synthetic Data

To reproduce the experimental setup in digital cameras, we generate two images by applying two different white balancing parameters both with the same U corresponding to the ones employed in Canon EOS Digital Rebel (shown in the Figure 6.6(b)). The diagonal values of the matrix Λ for the first image are chosen to be equal to $\{1.436, 1, 1.763\}$ and as $\{2.442, 1, 1.073\}$ for the second image. These values correspond to the ones used for *daylight* and *tungsten light* settings respectively. The coefficients of $A_{1\to 2} = U^{-1}(\Lambda^{(2)}/\Lambda^{(1)})U$ and the transformation matrix

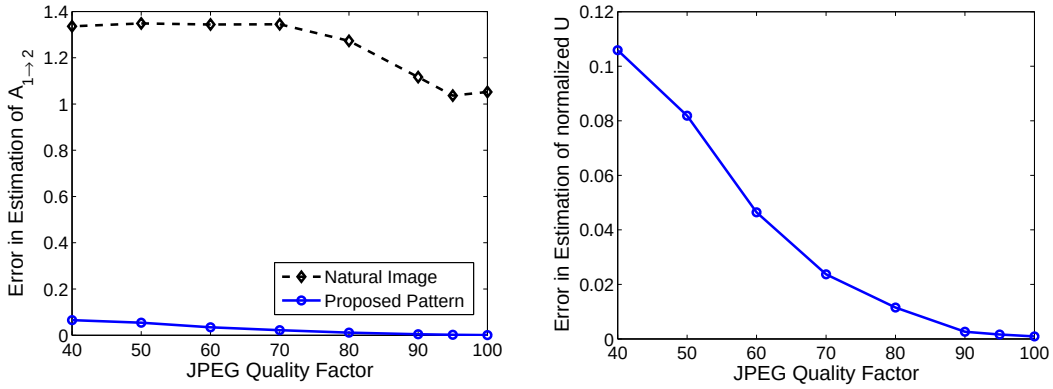


Figure 6.7: Results for white balancing showing the error in estimation of (a) $A_{1\to 2}$ and (b) normalized transformation matrix U_{norm}.

U are then estimated from these two white balanced images.

We study the robustness of the estimation techniques as the final images are JPEG compressed. More specifically, we JPEG compress the white balanced images with different quality factors and use these images for estimation. The estimation error in $A_{1\to 2}$ is computed as the squared Forbenius norm between the actual and the estimated values, and is shown in Figure 6.7(a) as a function of the JPEG quality factor. The figure shows the error for the synthetic pattern alongside the average error recorded from 20 natural images. We notice that the error reduces as the quality factor increases for both natural images and the designed pattern as expected. We also observe that the overall value of error for the designed pattern is an order of magnitude lower than that obtained for natural images. This result demonstrates the superiority of the proposed heuristic pattern for semi non-intrusive estimation of white balancing parameters.

Eigen value decomposition is applied to the estimated matrix $A_{1\to 2}$, and the eigenvector matrix $\hat{U}_{\text{norm}}$ is computed with each of the eigenvectors normalized to unit energy. The Frobenius norm between the actual normalized matrix U_{norm} and

the estimated matrix is shown in Figure 6.7(b) as a function of the JPEG quality factor. We notice that error values are lower than 0.1, suggesting the effectiveness of the proposed heuristic pattern for estimating the white balance parameter U_{norm}. Similar results were also obtained when tested with camera data.

Comparing Figure 6.4(a) and Figure 6.7(a), we also find that while the estimation results obtained in the semi non-intrusive scenario with the proposed heuristic pattern are better than the ones obtained using natural images in both cases, the performance improvement is more significant in the case of white balancing than for the case of color interpolation. This result can be attributed to the multiplicative nature of the white balance operation (see (6.16)), that requires more information to produce more accurate estimates, and such additional information may be available in controlled test conditions in a semi non-intrusive scenario. These results also suggest that the performance improvements obtained with semi non-intrusive forensics depends on the nature of processing that is to be identified.

Testing with Camera Images

We use the proposed estimation techniques for obtaining the white balancing parameters from camera data. In our experiments, we display the pattern in the Liquid Crystal Display (LCD) monitor and capture it with several digital cameras. All images are captured under the same constant uniform illumination under incandescent lights. The Gamma of monitor is set to 1 and the ISO setting and focal length are maintained to be similar for all images. A tripod is used to remove the effects of other kinds of such random distortions as the ones introduced by hand shaking, and distinct horizontal (and vertical) lines as shown in Figure 6.3 is used as a reference lines and to fix the center of the camera to the center of the

$$\hat{U}_{\text{norm}} = \begin{bmatrix} 0.9603 & -0.0520 & 0.0321 \\ -0.2777 & 0.8606 & -0.1233 \\ 0.0267 & -0.5065 & 0.9919 \end{bmatrix}$$

$$U_{\text{norm}} = \begin{bmatrix} 0.9977 & -0.3838 & 0.0371 \\ -0.0525 & 0.8794 & -0.2710 \\ 0.0437 & -0.2817 & 0.9619 \end{bmatrix}$$

Figure 6.8: Results for estimating white balancing parameters for Canon EOS Digital Rebel: the estimated and the actual values of the normalized transformation matrix (U_{norm}) are shown alongside for comparison.

image. Several snapshots of the input image were taken by changing the white balance setting on the camera manually.

As a preliminary pre-processing step, registration is performed on the two JPEG images. The corners in checker-board registration pattern is employed to give good set of corresponding points, and the homographies [57] are computed by matching these corners. One of the two images is then projected using the estimated homography and the projected image is used for subsequent analysis. We formulate a set of linear equations using the projected image and solve it using the least squares technique to obtain $A_{1\to 2}$. We then compute its eigenvalues and normalized eigenvector matrix $\hat{U}_{\text{norm}}$. The estimated values are shown in Figure 6.8. The actual value of the transformation matrix U is also obtained by reading the header of the corresponding raw files captured solely for testing purposes. The closeness in the estimated and the normalized actual coefficients demonstrate that our proposed simulation setting, pattern, and the estimation technique is good.

6.4 Optimal Pattern Design for Semi Non-Intrusive Forensics

In this section, we employ metrics from the estimation and pattern classification frameworks presented in Chapter 5 to optimize the heuristic pattern for semi non-intrusive analysis.

6.4.1 Optimizing the Heuristic Pattern via Estimation Framework

We optimize the input pattern for semi non-intrusive forensics by solving a minimization problem that minimizes the parameter estimation accuracies, $\Delta_s(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are the input and the output to the component, respectively. As described in the imaging model in Section 6.1, color interpolation in digital cameras can be expressed mathematically as (6.1) and can be represented in the matrix form as

$$\mathbf{y} = \mathbf{X}\underline{\alpha} + \mathbf{n}_1. \tag{6.18}$$

where $\mathbf{y}$ denotes the component output, $\mathbf{X}$ represents a matrix with component input values, $\underline{\alpha} = [\alpha(-N_\alpha, -N_\alpha, 1), \ldots, \alpha(N_\alpha, N_\alpha, 3)]^T$ is a vector containing all the component parameters to be estimated, and $\mathbf{n}_1$ is the additive white noise that models any post-interpolation processing operations. Under the assumption that the noise follows an independent and identically distributed Gaussian distribution, it can be shown that the estimation error for semi non-intrusive forensics under the input x is equal to the inverse of the signal-to-noise ratio, *i.e.*,

$$\Delta_s(\mathbf{x}, \mathbf{y}) = \sigma_{n_1}^2 (\mathbf{X}^T \mathbf{X})^{-1}. \tag{6.19}$$

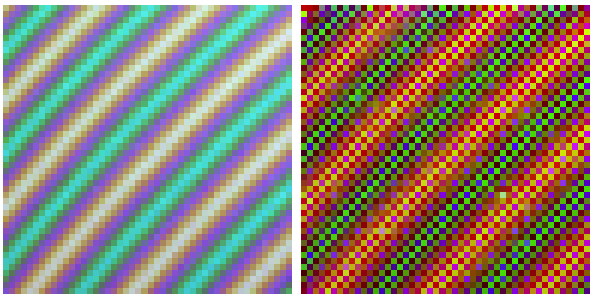


Figure 6.9: Digitally magnified versions of a 32×32 part in the original and optimized input patterns.

where $\sigma_{n_1}^2$ is the variance of the additive noise.

An iterative technique based on gradient-descent algorithm is employed to minimize the cost function $\Delta_s(\mathbf{x}, \mathbf{y})$ and to optimize the pixel values of the input pattern [126]. In Figure 6.9, we show the results of the optimization algorithm for a 32×32 part the original input along with the optimized version for comparison. To test the goodness of the designed pattern and the optimized pattern for estimating the cameras' color interpolation parameters, we first interpolate both the original and the optimized images shown in Figure 6.9 using different kinds of adaptive interpolation algorithms such as gradient based [92] and adaptive color plane [56]. We then post-process the interpolated images by JPEG compressing them under different quality factors [126]; and finally re-estimate the interpolation coefficients from the compressed versions. Figure 6.10 shows the estimation error as a function of the JPEG quality factor for both the heuristically designed input and the optimized input image. The figure shows the average error is significantly lower for the case of the optimized pattern compared with the original pattern. This result suggests that the theoretical framework can be employed to design optimal input patterns for estimating the color interpolation parameters with improved efficiency and robustness to post-interpolation operations such as JPEG compression.

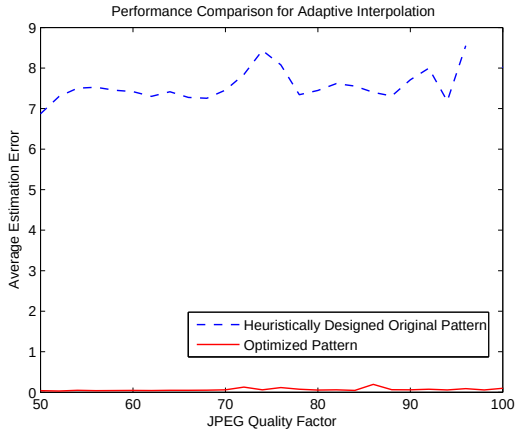


Figure 6.10: Average estimation error for semi non-intrusive forensics as a function of JPEG quality factor.

6.4.2 Optimizing the Heuristic Pattern via Pattern Classification Framework

In the previous subsection, we employed estimation error as a metric for optimizing the heuristic pattern for semi non-intrusive analysis. The optimized images obtained therein can be employed to estimate the coefficients with a higher accuracy as shown in the experimental results; and can be widely deployed for forensic analysis when the knowledge of possible set of color interpolation algorithms is not known apriori.

In this subsection, we show that with the knowledge of the possible set of color interpolation algorithms, ideas from pattern classification theory can be employed for optimizing the heuristic pattern and find the one that maximizes the overall confidence in decision making. As shown earlier in Section 6.3.1, the *optimal* input for camera component forensics is the one that maximizes the distance, $||(A_{\alpha\beta}(i) - A_{\alpha\beta}(j))\mathbf{x}||$. Here, $A_{\alpha\beta}(i)$ and $A_{\alpha\beta}(j)$ correspond to two different possible values for the $A_{\alpha\beta}$ from the algorithm space. It can be shown that the solution for this

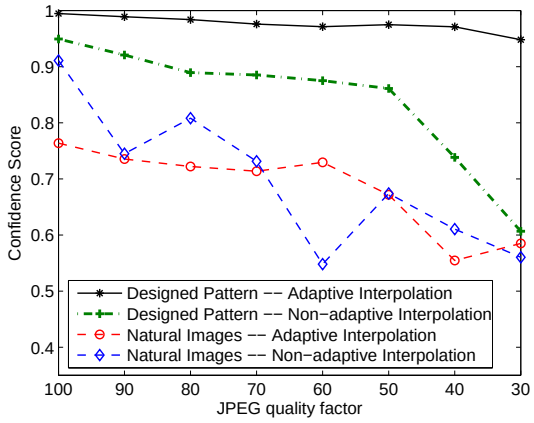


Figure 6.11: Confidence score as a function of JPEG quality factor for (a) natural images (b) designed pattern.

maximization problem, $\hat{\mathbf{x}}$, is along the direction of the eigenvector corresponding to the largest eigenvalue of the matrix $(A_{\alpha\beta}(i) - A_{\alpha\beta}(j))$. Based on the above observations, we optimize the heuristic pattern in Figure 6.3 and modify in such a way that the optimal input approximately follows the direction of the maximum eigenvector [130, 133]. The optimized input is employed for testing.

To simulate the camera capture process, the optimized input image is interpolated using two different interpolation techniques: bicubic that does not adapt to image content, and the adaptive color plane interpolation method (see Appendix I of Chapter 3 for detailed description of interpolation algorithms) that adapts to image gradient values. The interpolation coefficients are estimated and used as an input to a two-class support vector machine (SVM) classifier [148] for identification. This SVM has been trained with the coefficients obtained from natural images correspondingly interpolated with each of the same two different techniques. We study the robustness in parameter estimation under JPEG compression. In Figure 6.11, we plot the confidence values obtained on classification under different

quality levels of JPEG compression both for the designed pattern and for natural images. We notice from the figure that as the JPEG quality factor reduces and compression noise becomes stronger, the confidence of correctly identifying the interpolation coefficients reduces. Additionally, we observe that the confidence score obtained with the designed pattern is higher than the average scores obtained with natural images; demonstrating the superiority of designed pattern for semi non-intrusive analysis.

6.5 Chapter Summary

In this chapter, we present several applications of the theoretical framework and show its applicability for semi non-intrusive component forensics of digital cameras. We present case studies to examine digital camera components and theoretically derive the requirements for intrusive, semi non-intrusive, and completely non-intrusive forensics of digital camera components. Motivated by the conclusions from the theoretical analysis, we identify the basic requirements of a good input pattern for semi non-intrusive forensics, and construct an input pattern satisfying these conditions. We present a systematic methodology to estimate the parameters of the cameras' color interpolation and white balancing algorithms, and show through simulations that the proposed heuristic input pattern in controlled testing conditions provides an overall higher accuracy in parameter estimation. Comparisons with natural images obtained under non-intrusive forensic conditions suggest the need for robust semi non-intrusive forensics, and the superiority of the heuristic input pattern for parameter estimation. We then apply the theoretical framework to optimize the input pattern using estimation error and confidence score as metrics; and show that the accuracy in parameter identification can be improved via

such an approach. The features obtained from semi non-intrusive analysis provide useful evidence to analyze infringement/licensing, to construct good training sets for camera identification, and to provide ground-truth information for tampering detection.

Appendix: Brief Survey of Some Popular White Balancing Algorithms

There are many algorithms for white balancing [32]. In manual white balancing techniques, the scale factors are chosen based on the chosen illuminance options such as tube light, sunlight, incandescent lamps, cloudy lights, night vision, etc. On the other hand, auto white balancing algorithms compute these values from the picture based on estimate of the illuminance of the scene [12, 140]. Auto white balancing can be very broadly classified into three main categories based on their inherent assumptions - gray world, white patch, and retinex methods. The Gray world techniques work by assuming that average of all the pixel values in the world is gray. These techniques find the the scale factors by normalizing with respect to the mean of the image, mean of all images in the database, weighted mean of the image, or by using the image mean after truncation [49]. The white patch algorithms on the other hand assume that the maximum value of the scene is white, and normalize the pixel values to achieve it. Retinex methods are one of the oldest known techniques [18]. In this case, a path is first chosen, and the ratio between the pixel values to the maximum in the path is computed. Such process is repeated for many possible paths and the average ratio is found to be used as a normalization factor.

Chapter 7

Extrinsic Fingerprinting via Robust and Secure Image Hashing

In the previous chapters, we discussed forensic approaches to image authentication. In addition to these methods, when the original image is available at hand, traditional techniques based on cryptography and watermarking can also be employed to authenticate multimedia, verify content integrity, and prevent forgery [24, 28, 29, 146]. In this chapter, we focus on addressing the problem of multimedia forensics via *extrinsic fingerprinting*. Extrinsic fingerprints are external signals that are added to the image by the device after the image has been captured. These external signals can then be used to establish the authenticity of digital data and determine possible tampering. Compared with non-intrusive forensic analysis via intrinsic fingerprints, the use of extrinsic fingerprints necessitates the presence of the device at hand as the fingerprint needs to be added at the time of image acquisition. While this requirement imposes some additional

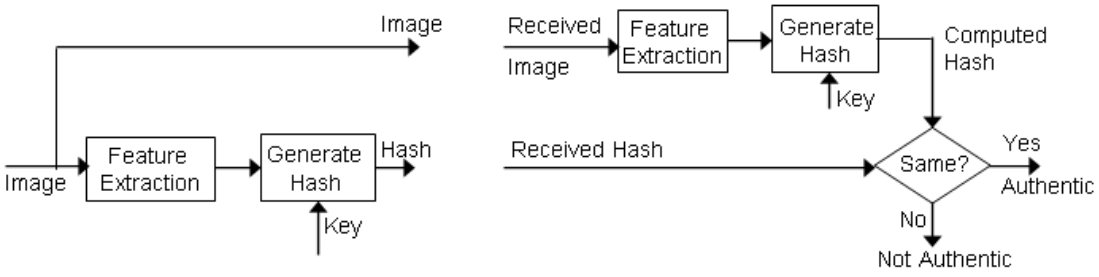


Figure 7.1: Hash functions for image authentication.

constraints on their applicability, extrinsic fingerprinting techniques help build an content-based image authentication scheme that is collision-resistant, robust to common signal processing operations, and secure against estimation and forgery attacks as will be shown in the chapter.

There are two popular approaches to multimedia authentication via extrinsic fingerprinting. These include semi-fragile watermarking [40, 52, 146] and robust image hashing [122]. In this work, we mainly focus on robust image hash functions as a means for extrinsic fingerprinting. A multimedia hash is a content-based digital signature of the media data. To generate a multimedia hash, a secret key is used to extract certain features from the data. These features are further processed to form the hash. The hash is transmitted along with the media either by appending or embedding it to the primary media data. At the receiver side, the authenticator uses the same key to generate the hash values, which are compared to the ones transmitted along with the data for verifying its authenticity. This process is illustrated in Figure 7.1.

In addition to content authentication, multimedia hashes are used in content based retrieval from databases [82]. To search for multimedia content, naïve methods such as sample-by-sample comparisons are computationally inefficient. More-

over, these methods compare the lowest level of content representation and do not offer robustness in such situations as geometric distortions. Robust image hash functions can be used to address this problem [138]. A hash is computed for every data entry in the database and stored with the original data in the form of a look-up table. To search for a given query in the database, its hash is computed and compared with the hashes in the look-up table. The data entry corresponding to the closest match, in terms of certain hash-domain distance that often accounts for content similarity, is then fetched. Since the hash has much smaller size with respect to the original media, matching the hash values is computationally more efficient.

Image hash functions have also been used in applications involving image and video watermarking. In non-oblivious image watermarking, the need for the original image in watermark extraction can be substituted by using hash as side information [20, 28, 30]. The hash functions have also been used as image-dependent keys for watermarking [41, 62]. In video watermarking, it has been shown that adversaries can employ "collusion attacks" to devise simple statistical measures to estimate the watermark if they have the access to multiple copies of similar frames [119]. A solution to this problem is to use secure, content-dependent hash values as a key to generate the watermark [42].

The rest of the chapter is organized as follows. In Section 7.1, we introduce the general framework for image hashing and present prior art. We then present the proposed image hashing scheme and compare its performance with several existing schemes in Section 7.2. We evaluate the security for a number image hashing schemes in Section 7.3. Finally, discussions are provided in Section 7.4 and the chapter is summarized in Section 7.5.

7.1 General Framework and Prior Art

There are two important design criteria for image hash functions, namely, *robustness* and *security* [42, 120–122, 138, 147]. By robustness, we mean that when the same key is used, perceptually similar images should produce similar hashes. Here, the similarity of hashes is measured in terms of some distance metric, such as the Euclidean or Hamming distance. In this work, we consider two images to be similar if one image can be obtained from the other through a set of content-preserving manipulations. This set of manipulations includes moderate levels of additive noise, JPEG compression, geometric distortions (such as the common rotation, scaling, and translation operations, or more generally affine transformations), cropping, filtering operations (such as spatial averaging and median filtering), and watermark embedding.

The security of image hash functions is introduced by incorporating a secret key in generating the hash. Without the knowledge of the key, the hash values should not be easily forged or estimated. Additionally, some design criteria for generic data hash also applies to image hash functions, namely, the one-way and collision-free properties. A hash is *one-way* if given a hash h and a hash function $g(\cdot)$, it is computationally expensive to find an image I such that $h = g(I)$. Collision-free property refers to the fact that given an image I and a hash function $g(\cdot)$, it is computationally hard to find a second image $\hat{I}$ such that $g(I) = g(\hat{I})$. Although some generic data hash functions such as MD5 satisfy these criteria [96], they are highly dependent on every bit (or pixel) of the input data rather than on the content. Hence, most of the them are not suitable for the emerging multimedia applications and the need for building robust and secure image hash is paramount.

To achieve robustness and security in image hashing, most of the existing

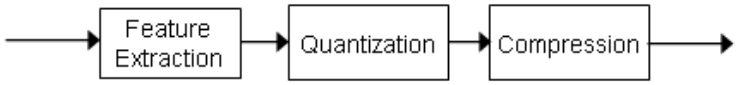


Figure 7.2: The three-step framework for generating a hash.

schemes follow a three-step framework to generate a hash. As shown in Figure 7.2, these three steps include

1. Generating a key-dependent feature vector from the image,

2. Quantizing the feature vector, and

3. Compressing the quantized vector.

The most challenging part of this framework has been the feature extraction stage [79, 98, 138]. A robust image feature extraction scheme should withstand minor distortions to the image that do not alter the semantic content [41, 41, 42, 78, 79, 98, 100, 120, 122, 138, 147, 151]. A typical approach is to extract image features that is invariant to allowed content-preserving image processing operations [41, 42, 78, 100, 151]. These features are then used to generate the hash values. Some of the features that have been proposed in the literature include block-based histograms [38, 64, 117], image edge information [115], relative magnitudes of the DCT coefficients [80], and the scale interaction model with the Mexican-Hat wavelets [16]. However, these features are both sensitive and publicly known. The sensitivity against minor distortion can be mitigated by preprocessing signals via low-pass filtering [138], applying quantization or extracting most-significant bits [151], and clustering [99]. As these resilient features are publicly known, using them alone makes the scheme susceptible to forgery attacks [42], even when the final hash is obtained by encrypting these features [16, 80]. This is because

the attacker may create a new image with different visual content, while still preserving the feature values. As the resulting hash will be the same, such hashing approaches may lead to mis-classifications in database applications, and would also be vulnerable to counterfeiting attacks in authentication applications. Therefore, the security mechanism should be combined into the feature extraction stage.

By jointly considering security and robustness, Fridrich *et al.* propose to generate image hash by projecting an input image onto zero-mean random smooth patterns, generated using a secret key [42]. While the resulting hash is resilient to filtering operations, it does not perform very well for geometric distortions and is not collision-free as shown in [116]. In [138], Venkatesan *et al.* use the principal values calculated from the wavelet transform of the image blocks to generate a feature vector invariant to general gray scale operations. The resulting features are then randomly quantized and compressed to produce the final hash [97]. Recently, it has been shown that this scheme does not perform well for some manipulations such as contrast changes, gamma correction [95]. An iterative key-dependent image hash based on repeated thresholding and spatial filtering was proposed in [98]. All these algorithms [42,98,138] described above perform well under additive noise and common filtering operations, but not under desynchronization and geometric distortions. Considering these disadvantages, the Radon soft hash algorithm (RASH) based on the properties of the Radon transform was proposed in [78,79]. Recently, other transform domain features have been employed for perceptual hashing. Features obtained from the singular value decomposition (SVD) of pseudo-randomly chosen regions of the image [71] and Randlet transform coefficients [90] have been shown to have good robustness properties especially for rotation and cropping attacks.

To enable fast comparison and searches, it is usually preferred that the final hash be a short sequence of bits rather than a set of real numbers. Therefore, the output of the feature extraction stage is usually quantized, converted to binary representation, and further compressed. Uniform, Lloyd-Max, or key-dependent randomized quantizers have been used for hash quantization [97, 138]; and the decoding stages of error correcting codes have been used for compressing the quantized hash [17, 97, 138]. These methods reduce the length of the hash vector; yet preserving the Hamming distance. Some work also secure the compression stage by performing a key-dependent random selection from the quantized hash values [97, 151]. A detailed survey of image hashing algorithms can be found in [147].

In this work, we introduce a new method to construct robust and secure image hash functions. Since the feature extraction stage is the most important stage in the general image hashing framework, we will investigate the feature extraction stage in greater detail in this chapter. We design a randomized hashing scheme based on the rotation invariance of the Fourier-Mellin transform. We show that the proposed scheme is robust to geometric distortions, filtering operations, and various content-preserving manipulations. We then present a framework to systematically study the security aspects of existing image hashing schemes. We propose to evaluate the security from an information theoretic perspective by measuring the amount of randomness in the hash vector using the differential entropy as a metric. We show that the suggested security evaluation framework is generic and can be used to analyze and compare the security of several classes of image hashing algorithms. We derive analytical expressions of security using an entropy-based metric for several representative image hashing schemes and demonstrate that the proposed hashing algorithm is more secure in terms of this metric. Finally, we

use the proposed security metric to discuss the trade-offs between robustness and security that is exhibited in most existing image hashing algorithms.

7.2 Image Hashing Algorithms Based on Polar Fourier Transform

In this section, we present the proposed image hashing algorithm [122]. Our proposed scheme is based on the Fourier-Mellin transform, which has been shown to be invariant to 2D affine transformations [41, 63, 81, 105]. We incorporate key-dependent randomization into the Fourier transform outputs to form secure and robust image hash.

7.2.1 Underlying Robustness Principle of the Proposed Algorithm

Consider an image $i(x, y)$ and its 2D Fourier transform $I(f_x, f_y)$, where f_x and f_y are the normalized spatial frequencies in the range $[0, 1]$. We denote a rotated, scaled and translated version of the $i(x, y)$ as $i'(x, y)$. We can relate them as

$$i'(x, y) = i(\sigma(xcos\alpha + ysin\alpha) - x_0, \sigma(-xsin\alpha + ycos\alpha) - y_0), \tag{7.1}$$

where the rotation, scaling, and translation (RST) parameters are α, σ, and (x_0, y_0) respectively. The magnitude of the 2D Fourier transform of $i'(x, y)$ can be written as

$$|I'(f_x, f_y)| = |\sigma|^{-2}|I(\sigma^{-1}(f_xcos\alpha + f_ysin\alpha), \sigma^{-1}(-f_xsin\alpha + f_ycos\alpha))|. \tag{7.2}$$

Consider now a polar coordinate representation in the Fourier transform domain, i.e. $f_x = \rho cos\theta$ and $f_y = \rho sin\theta$, where $\rho \in [0, 1]$ is the normalized radius and

$\theta \in [0, 2\pi)$ is the angle parameter. The (7.2) can be written using polar co-ordinates as

$$|I'(\rho, \theta)| = |\sigma|^{-2} |I(\rho\sigma^{-1}, \theta - \alpha)|. \tag{7.3}$$

In (7.3), we observe that the magnitude of the Fourier transform is independent of the translational parameters (x_0, y_0). Observing that a rotation in image domain leads to a rotation by the same amount in the Fourier transform domain, we integrate the transform magnitude $|I'(\rho, \theta)|$ along a circle centered at zero frequency with a fixed radius ρ to obtain

$$h(\rho) = \int_0^{2\pi} |I'(\rho, \theta)| d\theta \approx \int_0^{2\pi} |I(\rho, \theta - \alpha)| d\theta \approx \int_0^{2\pi} |I(\rho, \theta)| d\theta. \tag{7.4}$$

These properties of the Fourier transform enable us to construct robust features. In the next subsection, we present the detail steps of the proposed algorithms.

7.2.2 Basic Steps of the Proposed Algorithms

The basic steps of the proposed algorithm include preprocessing, feature generation, and post processing.

1. *Preprocessing:* We first apply a low-pass filter on the input image and down-sample it. We then perform histogram equalization on the down-sampled image to get $i(x, y)$. We take a Fourier transform on the preprocessed image to obtain $I(f_x, f_y)$. The Fourier transform output is converted into polar co-ordinates to arrive at $I'(\rho, \theta)$ as in (7.3).

2. *Feature generation:* We sum up $I'(\rho, \theta)$ along the θ-axis at K equidistant points in the range of $[0, 2\pi)$, i.e. for $\theta \in \{\frac{\pi}{K}, \frac{3\pi}{K}, \ldots, \frac{(2K-1)\pi}{K}\}$, to obtain an image feature vector h_ρ. $K = 360$ is used in our implementation. Since the

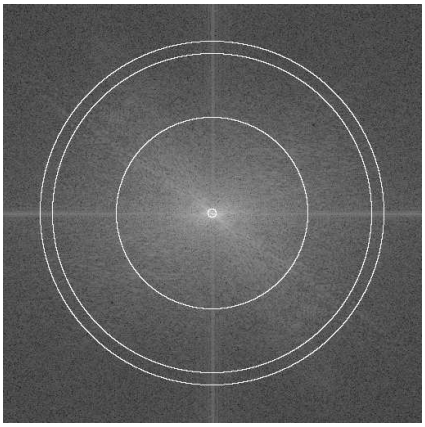


Figure 7.3: 2-D Fourier transform of the Lena image. The j^{th} hash value$-h_j$, is obtained by a random weighted summation along the circumference of chosen radii $\rho \in \Gamma_j$ in scheme-2. Some of the constant radii circles used in the summation are displayed in the figure. The magnitude of the Fourier transform is shown in the log-scale and has been appropriately scaled for display purposes.

feature h_ρ is only dependent on the image content, we propose two randomization methods to obtain key-dependent features using h_ρ:

- **Scheme 1:**

  We obtain $|I'(\rho, \theta)|$ as in (7.3) and compute a weighted sum along the θ-axis to obtain the j^{th} hash value:

  $$h_j = \sum_{i=0}^{K-1} \beta_{\rho_j, i} \left| I' \left(\rho_j, \frac{(2i+1)\pi}{K} \right) \right|, \tag{7.5}$$

  where $\{\beta_{\rho_j, i}\}$ are key-dependent pseudo-random numbers that are normally distributed with mean m and variance σ^2.

- **Scheme 2:**

  We first use a secret key to generate random sets of radii $\{\Gamma_j\}$. We then take $|I'(\rho, \theta)|$ obtained in (7.3) and do a summation along the θ-axis

for each radii in this set. A random linear combination of the resulting summations gives the j^{th} hash value. This can be represented as

$$h_j = \sum_{\rho \in \Gamma_j} \beta_\rho \sum_{i=0}^{K-1} \left| I' \left(\rho, \frac{(2i+1)\pi}{K} \right) \right|, \tag{7.6}$$

where β_ρ are key-dependent pseudo-random numbers that are normally distributed with mean m and variance σ^2. This method is illustrated in Figure 7.3.

3. *Post processing:* We quantize the resulting statistics vector and apply Gray coding to obtain the binary hash sequence [51]. This bit sequence is then passed through the decoding stage of a order-3 *Reed-Muller* decoder for compression [97]. This step may also be replaced with the *Wyner-Ziv* encoder [65, 149]. Furthermore, we can enhance the security of the hash by making the quantization and compression stages key-dependent. For example, randomized quantization algorithms may be used to quantize the hash [97]; for the compression stage, we can randomly select the hash values from the quantized hash vector [98] or randomly choose the order of the *Reed-Muller* decoder used for different sub-sections of the hash. These techniques would further enhance the security of the resultant hash vector. Finally, the compressed hash is randomly permuted according to a permutation table generated using the key.

7.2.3 Performance Study and Comparison

Performance Metrics and Experiment Setup

To measure the performance of image hashing, we choose the Hamming distance between the binary hashes, normalized with respect to the length (L) of the hash as a performance metric. The normalized Hamming distance is defined as

$$d(h_1, h_2) = \frac{1}{L} \sum_{k=1}^{L} |h_1(k) - h_2(k)|, \tag{7.7}$$

which is expected to be close to 0 for similar images and close to 0.5 for dissimilar ones. As more parts of a picture is changed, the manipulated image and the original image become more dissimilar. For an ideal hashing scheme, the normalized Hamming distance between the corresponding hashes should increase accordingly.

We test the proposed schemes on a database of around 157,200 images. In this database, there are 1200 original grey scale images each of size 512×512. This includes around 50 classic benchmark images (such as Lena, Baboon, Pepper, etc.), and a variety of scenery and human activity photos taken by digital cameras. These camera photos were cropped, converted to grey scale, and downsampled to 512×512. For each original image in this set, we generate 130 similar versions by manipulating the original image according to a set of content-preserving operations listed in Table 7.1. We measure the normalized Hamming distance between the hashes of the original image and the manipulated images. The results obtained for the proposed schemes are compared with three representative existing schemes by Fridrich *et al.* [42], by Venkatesan *et al.* [138], and by Mihçak *et al.* [98]. These three schemes are chosen because they adopt different ways to extract the robust image feature as well as different methods to randomize these features. We also consider the normalized Hamming distance between the hashes of dissimilar images, which indicates the discriminative capability of the hashing algorithm. We note that the computed hashes of all these schemes are short in length. For a 512×512 image,

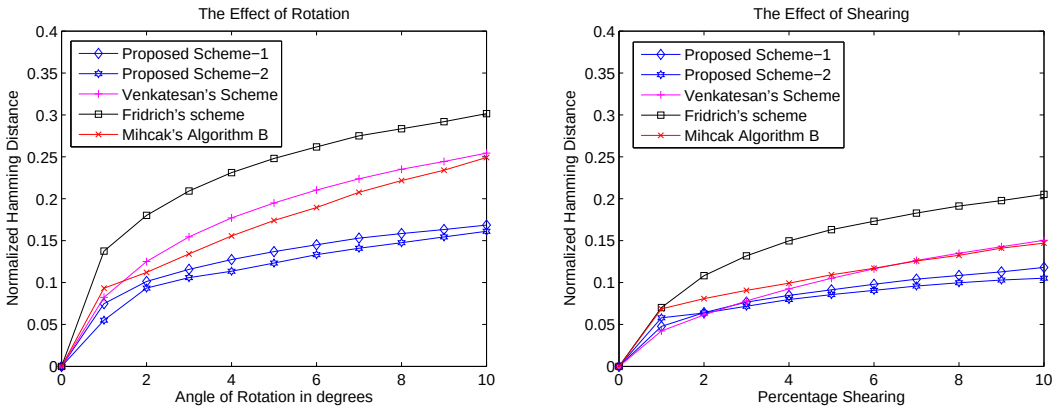


Figure 7.4: Performance of various hashing schemes under desynchronization attacks. To generate a point on the curve, the input image was first rotated (or sheared) to give a larger image padded appropriately with zeros. This image was then cropped to exclude the zeros and resized to a pre-determined canonical size. The hash of the resulting image was computed and the normalized Hamming distance from the hash of original image is shown in the Y-axis.

the hash lengths are on the order of a few hundred bits, as shown in Table 7.2.

Experimental Results on Robustness of the Hash

To examine the robustness properties, we consider the performance of various hashing schemes to different content-preserving manipulations such as moderate RST, filtering, and image compression.[1] We show the comparison results in terms of normalized Hamming distance in Figure 7.4−Figure 7.8. Our results indicate

[1]In all the experiments, we use our implementation of the hashing methods [42, 98, 138] for the comparison study. Whenever possible, we verified the performance results with the ones reported in the paper. In all cases, the parameters of the hashing algorithms were chosen so as to maintain similar values for the security metric in order to facilitate a fair comparison. Refer Section 7.3 for details on the security metric.

Table 7.1: Set of content-preserving manipulations.

Manipulation Operation	Parameters of the Operation	Number of Images
Additive Noise		
Gaussian distributed	Variance 0-0.2	10
Uniform distributed	Variance 0-0.5	10
Filtering Operations		
Spatial Averaging	Filter order 2-6	5
Median Filter	Filter order 2-11	10
Wiener Filter	Filter order 2-11	10
Sharpening	Filter order 3-11	5
Geometric Distortions		
Rotation	Degrees 1-20	20
Scaling	Percentage 0.5-1.5	10
Cropping	Percentage 1-30	10
Shearing	Percentage 1-10	10
Random deletion of lines	Percentage 1-20	10
Luminance Non-Linearities		
Gamma correction	$I^{\gamma}, \gamma \in$ [0.75-1.25]	10
JPEG compression	Compression Ratio 10-99	10
Total		130

Table 7.2: Hash lengths for various hashing schemes.

Hashing method used	Hash Length
Mihçak's algorithm B [98]	1000
Venkatesan's scheme [138]	805
Fridrich's scheme [42]	420
Proposed scheme 1	420
Proposed scheme 2	420

that the proposed schemes perform well under desynchronization distortions. The performance for rotation and shearing distortions, averaged over the 1200 images, are shown in Figure 7.4. In the case of rotation distortions, we observe that the Hamming distance between the quantized feature vectors of the proposed schemes is smaller than those of the existing schemes, especially for large rotation angle. This is expected since the summation along the θ-axis reduces the effects of rotation. We can also observe that scheme-2 gives better results than scheme-1, in terms of the normalized Hamming distance. This is attributed to the fact that performing a weighted sum along the θ-axis as in the proposed scheme-1 no longer preserves rotation invariance. The proposed algorithms also achieve comparable performance with most existing algorithms under shearing distortions. The performance results for random bending [107] and cropping are shown in Figure 7.5(a) and (b) respectively. We observe that the proposed schemes perform very well for both these distortions. This is because the magnitude of the low frequency coefficients of the Fourier transform that contribute to the hash does not change much under moderate bending and cropping.

We show the performance of the hash algorithms under additive noise in Fig-

Table 7.3: Performance of the algorithm for dissimilar images under the type of manipulation shown in Figure 7.9. Here, d_{AB} denotes the distance between images (a) and (b).

Hashing method used	d_{AB}	d_{AC}	d_{BC}
Mihçak's algorithm B [98]	0.50	0.20	0.28
Venkatesan's scheme [138]	0.37	0.15	0.31
Fridrich's scheme [42]	0.41	0.26	0.34
Proposed scheme 1	0.49	0.28	0.37
Proposed scheme 2	0.48	0.32	0.39

ure 7.6. We observe from the figure that the proposed scheme−2 does well compared to the proposed scheme−1 and other existing schemes. We further note that the normalized Hamming distance between the hashes of the noisy image and the original image is very small and on the order of 0.02. This performance is attributed to the low pass filtering in the preprocessing step of the hash generation. The results for filtering and JPEG compression are shown in Figure 7.7 and Figure 7.8. We observe that the performance of the proposed schemes under these distortions is comparable to the existing schemes.

The Discriminative Capability of Hash

Since image hash should be able to distinguish malicious manipulations from content-preserving ones, its performance in differentiating images with different contents is an important performance aspect. For images with different contents, an ideal hash algorithm should produce two statistically independent binary hash vectors, where half of the hash bits are expected to be the distinct and the other

half the same. This would result in a normalized Hamming distance of around 0.5. Our experiments with a set of 1200 different images indicate that the mean of normalized Hamming distance of the resulting 719,400 combinations was around 0.48. To further demonstrate the performance of the proposed scheme to inauthentic modifications, we consider the following cut-and-paste image editing as shown in Figure 7.9, where a new image (c) is created by combining approximately equal parts from image (a) and (b). An ideal image hashing scheme should classify (c) as inauthentic. We perform this test on 500 images and list the normalized Hamming distance between the obtained hash vectors for different algorithms in Table 7.3. We can see from the table that the proposed schemes find the image (c) to have large distances from (a) and (b), and thus correctly declare it inauthentic; on the other hand, the existing algorithms suggest a smaller distance and have lower reliability to distinguish (c) from (a) and (b).

Image Authentication as a Hypothesis Testing Problem

Generally speaking, the problem of image authentication can be considered as a hypothesis testing problem with the following two hypotheses

- H_0: Image is not authentic; and
- H_1: Image is authentic.

Now, we examine the robustness and discriminative capabilities of various hashing schemes in terms of the Receiver Operating Characteristics (ROC) [109, 145]. The ROC curve characterizes the receiver's performance by classifying the received signal into one of the hypothesis states. For each original image, we compute and store the hash values, which we denote as h_1. Given the received image, we find its hash value h_2 and declare it to be authentic if the normalized Hamming distance

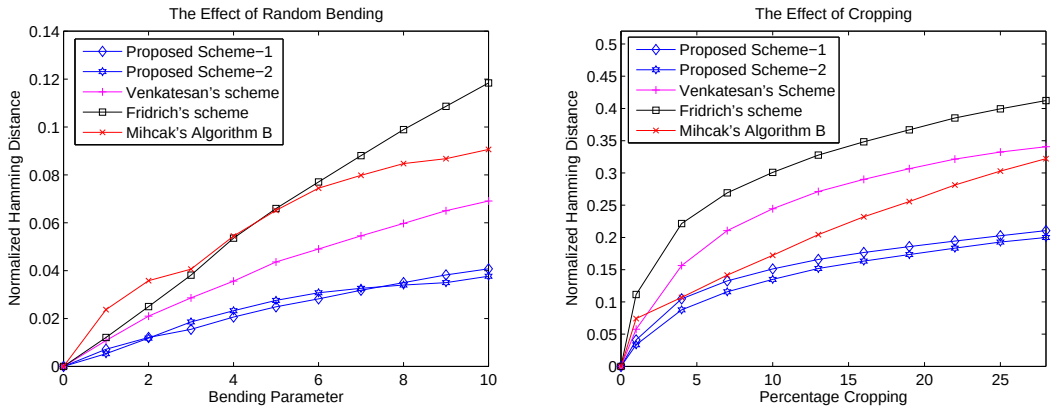


Figure 7.5: Performance of various hashing schemes under (a) bending and (b) cropping. Cropped images were obtained by retaining the central portion of the image and removing the boundaries. The cropped image is resized to a pre-determined canonical size before computing the hash.

between the hashes satisfies $d(h_1, h_2) < \eta$ where η is a decision threshold. Based on ground truth, we record the number that are correctly classified as authentic to give us an estimate of the probability of correct detection (P_D). For a given η, we also record the number of processed versions of other images that are falsely classified as original image and obtain an estimate of the probability of false alarm (P_F). We repeat this process for different decision thresholds η, and arrive at the ROC. The ROC obtained from the experiments using 1200 different images is shown in Figure 7.10. We can observe from the ROC curves that the proposed schemes attain a $P_D = 0.95$ when the P_F is 0.05, while the other schemes attain the same P_D when P_F is close to 0.15. Hence, the proposed scheme has a higher probability of correct detection for a given probability of false alarm and hence achieves a better performance. This further demonstrates the advantages of the proposed hashing schemes over the existing schemes.

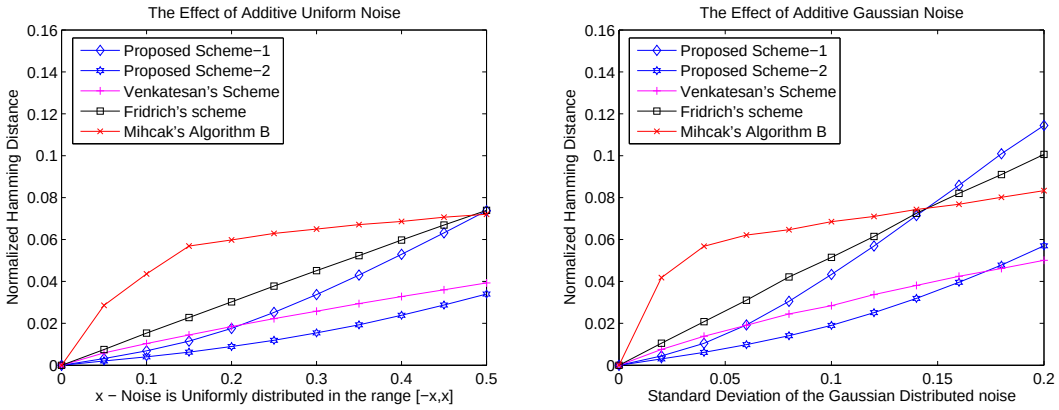


Figure 7.6: Performance of various hashing schemes under additive noise. The noisy images were artificially generated by adding uniform/Gaussian distributed noise of different variances to the original image.

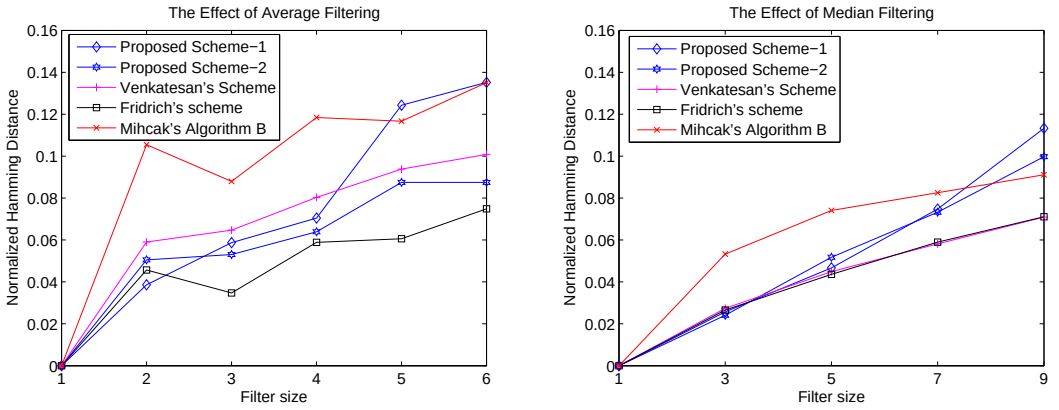


Figure 7.7: Performance of various hashing schemes under filtering.

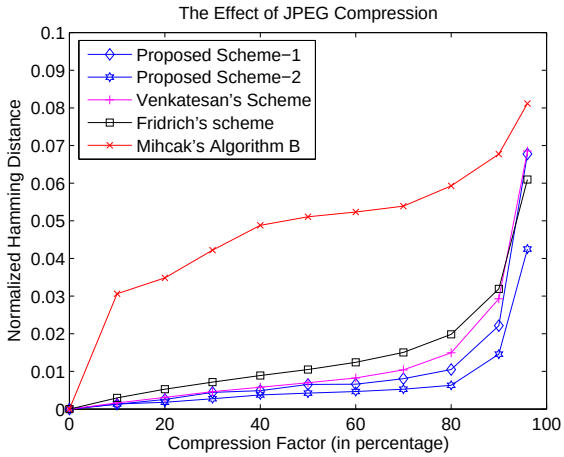


Figure 7.8: Performance of various hashing schemes under JPEG compression.

7.3 Security Analysis

In addition to robustness, another important performance aspect of image hashing is security, i.e. the hash values should not be easily forged or estimated without the knowledge of the secret key. In this section, we introduce a framework to evaluate and compare the security of image hashing schemes. We propose to use differential entropy as a metric to study the security of randomized image features and derive analytical expressions of the proposed metric for some representative classes of image hashing algorithms. Further extensions of the proposed framework and other possible approaches to study security are described later in Section 7.4.3.

7.3.1 The Proposed Security Evaluation Framework

We propose to evaluate the security of image hashing schemes from an adversary view point. The adversary knows the hashing algorithm $g(\cdot)$ and the image I, and tries to estimate the hash values without the knowledge of the secret key. The degree of success that can be attained by the adversary depends on the amount of

(a) (b) (c)

Figure 7.9: An example of inauthentic manipulations obtained by combining parts of multiple images. (a) and (b) are two original 512 × 512 images. Image (c) is obtained by combining parts of image (a) and (b).

randomness in the hash values. The higher the amount of randomness in the hash values, the tougher it would be to estimate or duplicate the hash without knowing the key. In the subsequent discussions, we shall focus on the security of the output of the feature extraction stage. Since the quantization and the compression stages are chained with feature extraction stage, once the entropy of this stage is obtained, the entropy measure for the following stages can be obtained subsequently.

We start the discussion by reviewing the definition of differential entropy [27]. The differential entropy of a continuous random variable X is denoted by $\aleph(X)$ and given by

$$\aleph(X) = \int_{\Omega} f(x) \log_2 \left(\frac{1}{f(x)} \right) dx \tag{7.8}$$

where $f(x)$ is the probability density function of X and Ω is the range of support of $f(x)$. In most image hashing schemes, the output of the feature extraction stage consists of two components – a deterministic part and a random part. The deterministic part is contributed by the image content, which we will consider to be known or can be well approximated from the test version of the image that the attacker can acquire. The random part is contributed by the pseudo-random

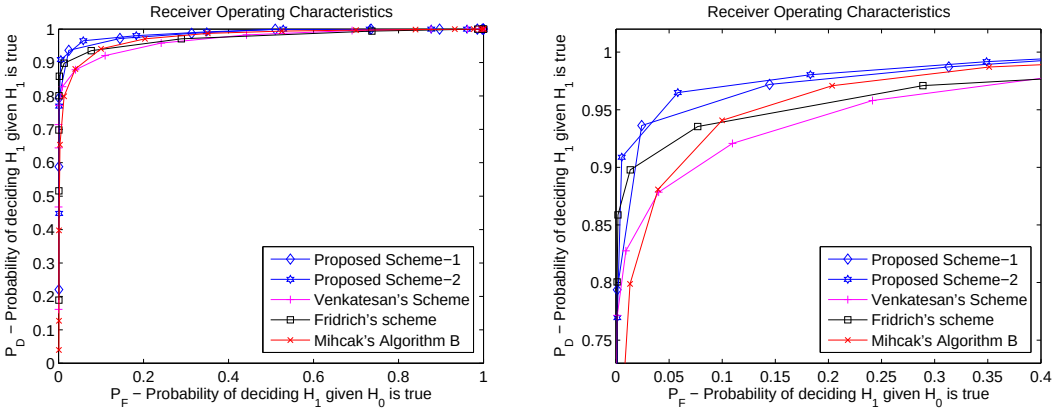


Figure 7.10: Receiver Operating Characteristics of the hypothesis testing problem. The plots display the probability of correct decision (P_D) with respect to the probability of false alarm (P_F). A greater the value of P_D for the same P_F indicates more robustness. The original curve is shown on the left and the magnified version is shown on the right.

numbers generated using the secret key. In our analysis, we model the output of the feature extraction stage as random variables and find the degree of uncertainty in terms of the differential entropy to arrive at the security metric [121]. In the following sections, we present the security analysis for our proposed scheme, and compare it with the results obtained for a number of representative prior work on image hashing [42, 98, 138].

7.3.2 Analytic Expressions of the Security Metric for the Proposed Schemes

In this part, we derive analytic expressions of the security metric for the proposed schemes. In the proposed scheme−1, the randomness in the hash is introduced by the variables $\{\beta_{\rho_k,i}\}$, which are key-dependent pseudo-random numbers, normally

distributed with mean m and variance σ^2. The final hash can be considered as a weighted summation of these Gaussian distributed random variables as shown in (7.5), where the weights of the summation are determined by the image content and known to the users. Since the sum of Gaussian random variables is also Gaussian, the hash value h_k will be Gaussian distributed with mean and variance given by

$$E(h_k) = m \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right|, \tag{7.9}$$

$$Var(h_k) = \sigma^2 \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right|^2. \tag{7.10}$$

Therefore, the differential entropy of the feature extraction stage for the proposed scheme-1 can be written as

$$\aleph(h_k) = \frac{1}{2} \log_2 \left((2\pi e)\sigma^2 \sum_{i=0}^{K-1} \left| I' \left(\rho_k, \frac{(2i+1)\pi}{K} \right) \right|^2 \right). \tag{7.11}$$

We observe that the differential entropy increases as the variance σ^2 becomes large and the scheme becomes more secure as expected. Additionally, we note that the differential entropy rises as the number of sample points K is increased. This is also expected since a higher value of K implies that we involve more random numbers for generating each hash value as shown in (7.5); and hence the hash would be more difficult to forge.

Next, we derive the security metric for the proposed scheme-2. In this scheme, we use the secret key to generate random sets of radii $\{\Gamma_k\}$, and the weights (β_ρ) for the summation in (7.6). To facilitate discussions, we define q_ρ as the summation of the polar Fourier transform coefficients at the radius ρ given by

$$q_\rho = \sum_{i=0}^{K-1} \left| I' \left(\rho, \frac{(2i+1)\pi}{K} \right) \right|. \tag{7.12}$$

The ρ values chosen for generating the hash are from $\Gamma_\rho = \{\rho_1, \rho_2, \ldots, \rho_N\}$. Let λ_{ik} be Bernoulli distributed random variables such that $P(\lambda_{ik} = 0) = P(\lambda_{ik} =$

$1) = 0.5$. We rewrite (7.6) in terms of q_ρ and λ_{ik} to obtain

$$h_k = \sum_{i=1}^{N} \lambda_{ik} \beta_{ik} q_{\rho_i}. \tag{7.13}$$

We observe that each hash value obtained is a weighted summation of N terms and each of these terms is a product of a Bernoulli and a Gaussian distributed random variable. Therefore, the hash value h_k is not Gaussian. To find the differential entropy of h_k, we first find the probability density function (pdf) of h_k using the (7.13) and then use the pdf to find the entropy. To derive the pdf, we compute the characteristic function of h_k and apply its inverse Fourier transform [106]. It can be shown that the pdf, $f_{h_k}(x)$, has a rather complicated form with 2^N terms and is given by

$$\begin{aligned} f_{h_k}(x) &= \frac{1}{2^N}\delta(x) + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}} \sum_{i=1}^{N} e^{-\frac{(x-mq_{\rho_i})^2}{2\sigma^2 q_{\rho_i}^2}} + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}} \sum_{i_1=1}^{N} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{N} e^{-\frac{(x-m(q_{\rho_{i_1}}+q_{\rho_{i_2}}))^2}{2\sigma^2(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2)}} \\ &+ \frac{1}{2^N}\frac{1}{\sqrt{2\pi}} \sum_{\substack{i_1,i_2,i_3=1 \\ i_1 \neq i_2 \neq i_3}}^{N} e^{-\frac{(x-m(q_{\rho_{i_1}}+q_{\rho i_2}+q_{\rho_{i_3}}))^2}{2\sigma^2(q_{\rho_{i_1}}^2+q_{\rho_{i_2}}^2+q_{\rho_{i_3}}^2)}} + \ldots + \frac{1}{2^N}\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-m\sum_{i=1}^{N} q_{\rho_i})^2}{2\sigma^2(\sum_{i=1}^{N} q_{\rho_i}^2)}}, \end{aligned} \tag{7.14}$$

where $\delta(\cdot)$ denotes the dirac delta function. We observe that the pdf of h_k is a sum of many Gaussian pdf's and finding the exact expression for the differential entropy by integrating (7.8) would not be feasible. We instead find the lower and upper bounds of the differential entropy. Using the concavity property of the entropy, we

arrive at a lower bound for the differential entropy

$$\begin{aligned}\aleph(h_k) \;\geq\; & \frac{1}{2^N}\sum_{i=1}^{N}\frac{1}{2}\log_2(2\pi e\sigma^2 q_{\rho_i}^2) + \frac{1}{2^N}\sum_{i_1=1}^{N}\sum_{\substack{i_2=1\\ i_1\neq i_2}}^{N}\frac{1}{2}\log_2(2\pi e\sigma^2(q_{\rho_{i_1}}^2 + q_{\rho_{i_2}}^2)) \\ & + \frac{1}{2^{N+1}}\sum_{\substack{i_1,i_2,i_3=1\\ i_1\neq i_2\neq i_3}}^{N}\log_2(2\pi e\sigma^2(q_{\rho_{i_1}}^2 + q_{\rho_{i_2}}^2 + q_{\rho_{i_3}}^2)) + \ldots \\ & + \frac{1}{2^{N+1}}\log_2\left(2\pi e\sum_{i=1}^{N}\sigma^2 q_{\rho_i}^2\right). \end{aligned} \tag{7.15}$$

This lower bound can be simplified using the following energy compaction property of the Fourier transform. Without any loss of generality, we assume that the radii are ordered as $\rho_1 < \rho_2 < \rho_3 < \ldots < \rho_N$. Now, since q_{ρ_i} is the summation of the absolute values of the Fourier transform coefficients along the circumference of the circle of radius ρ_i, we have

$$q_{\rho_1} \geq q_{\rho_2} \geq \ldots \geq q_{\rho_N} \tag{7.16}$$

for most natural images. Using this inequality, (7.15) can be simplified to give a compact lower bound

$$\aleph(h_k) \geq \frac{2^N - 1}{2^{N+1}}\log_2(2\pi e\,\sigma^2 q_N^2) + \frac{1}{2^N}\sum_{i=1}^{N}\binom{N}{i}\log_2(i). \tag{7.17}$$

Next, to derive the upper bound, we use the fact that the Gaussian distribution has the maximum differential entropy among all distributions with the same variance. Moreover, the differential entropy of a Gaussian distributed random variable depends only on its variance. Therefore, we obtain an upper bound on $\aleph(h_k)$ by finding variance of the hash values h_k, from the pdf. in (7.14), to arrive at

$$\aleph(h_k) \leq \frac{1}{2}\log_2\left((2\pi e)\left(\frac{\sigma^2}{2} + \frac{m^2}{4}\right)\sum_{j=1}^{N} q_{\rho_j}^2\right). \tag{7.18}$$

In Figure 7.11, we show the derived lower and upper bounds along with the actual value, for different number of sampling points (N). The true values were

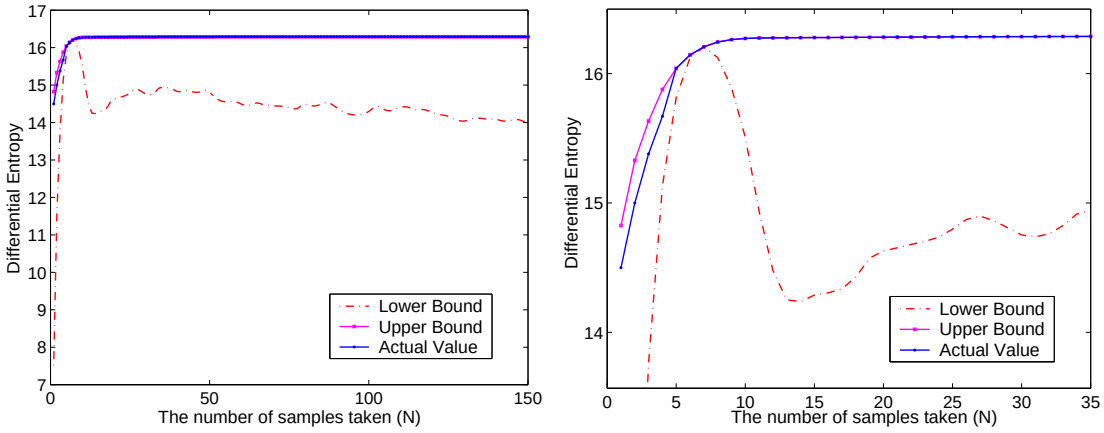


Figure 7.11: The entropy of the hash values for the proposed scheme−2 plotted with respect to the number of sampling points N. The plots show the lower bound, the upper bound and the actual value. The actual plot is shown on the left and the magnified version is shown on the right.

obtained by numerically computing the differential entropy from the pdf. of the hash values. We observe that the upper bound plotted using (7.18) is very tight and is almost equal to the actual value. This is because the true pdf. of the hash values is close to Gaussian with the same mean and variance as those used in the upper bound calculation.

7.3.3 Extending the Security Evaluation to Other Image Hashing Schemes

In this subsection, we show that the proposed security metric can be extended to study the security of various classes of image hashing schemes and is thus generally applicable. For our study, we consider two representative methods, namely, the scheme by Fridrich *et al.* [42] and the hashing algorithm by Venkatesan *et al.* [138]. These schemes were chosen as they have very different approaches to introduce

randomness in the feature extraction stage. For instance, the Fridrich's scheme [42] secures the hash by projecting the image onto random low-pass images; and the Venkatesan's scheme [138] introduces security by extracting image features from randomly chosen regions of the image.

Security of Fridrich's scheme [42]

This scheme is based on the observation that any significant change made in the transform domain would be reflected as visible changes in the image domain. Key-dependent pseudo-random patterns $\{X^{(r)}\}$, of the same size of the image, are initially generated. These patterns are then spatially averaged with a $m \times n$ low-pass filter $\{\alpha_{ij}\}$ to generate zero-mean smoothened random patterns $[Y^{(r)}]_{kl}$. The r^{th} hash value h_r is obtained by projecting the input image on to $Y^{(r)}$, as given by

$$h_r = \sum_{k=1}^{H} \sum_{l=1}^{W} Y_{kl}^{(r)} I_{kl}. \tag{7.19}$$

To analyze the security of this scheme, we consider the hash values $\{h_r\}$ as random variables and find their distributions. Using this estimated pdf, we compute the differential entropy as

$$\aleph(h_r) \approx \frac{1}{2} \log_2 \left(2\pi e \, \frac{1}{12} \sum_{p=1}^{H} \sum_{q=1}^{W} I_{pq} I_{pq}^{(\alpha\alpha)} \right). \tag{7.20}$$

Here, $I^{(\alpha\alpha)}$ is the image obtained by filtering I twice with the filter $\{\alpha_{ij}\}$. The details of the analysis is presented in Appendix I of this chapter.

Figure 7.12 shows the plot of the differential entropy of the Fridrich's scheme for different orders of averaging filter. We observe from the plot that the differential entropy decreases as the order of the filter is increased. This result is expected because on increasing the order of the averaging filter, the degree of uncertainty in the smoothened patterns $\{Y^{(r)}\}$ decreases, as the original random images $\{X^{(r)}\}$

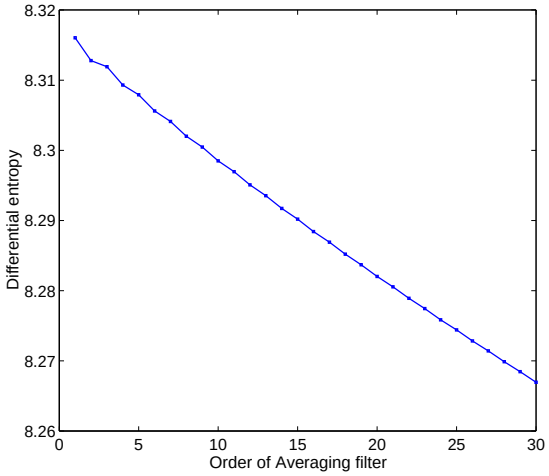


Figure 7.12: Differential entropy of the hash for different orders of averaging filters in Fridrich's scheme [42].

are low-pass filtered to a greater extent. Thus, the amount of randomness of the final hash values reduce as a consequence.

Security of Venkatesan's Scheme [138]

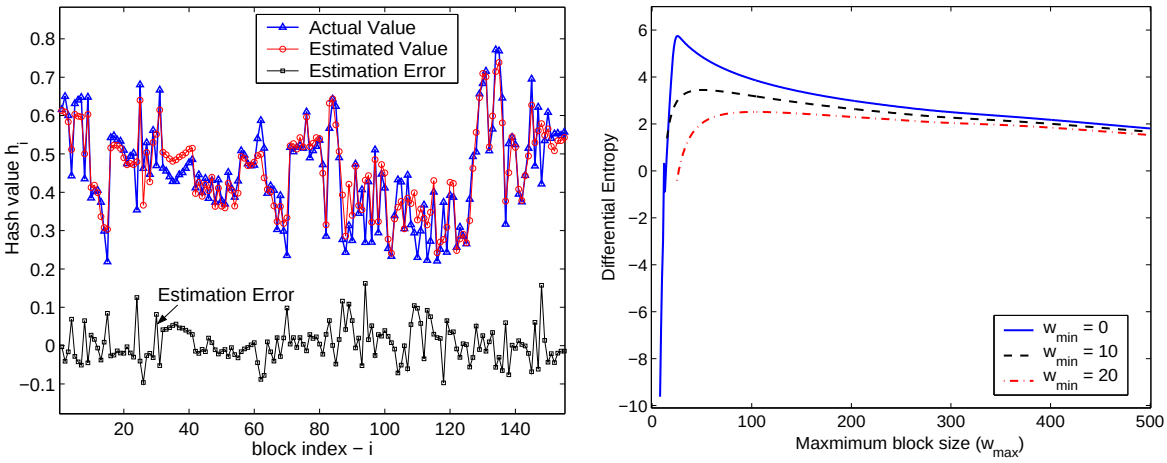
In this scheme, the authors first perform a 3-level DWT of the image and then a random tiling of each DWT sub-band of the image is generated. The mean (or variance) of the pixel values in the random rectangle is used to form the feature vectors [138]. These features are then randomly quantized and compressed to generate the hash.

There are two aspects of security in this scheme. To estimate the hash values, the adversary has to first find the locations and sizes of the random partitions and compute the image statistics in these partitions. Then, the adversary needs to arrange the estimated hash values in the correct order to obtain the hash vector. In our analysis, we consider these two aspects separately and obtain the differential

entropy in each case.

We first show that the exact size and location of the random partitions is not required to estimate the hash. The attacker can instead make an intelligent guess of the image statistics by replacing the random partitions with uniformly spaced, equal sized partitions. In [138], the width of the random partition is uniform in $[w_{min}, w_{max}]$, where w_{min} and w_{max} are the minimum and maximum widths of the random block. Therefore, a good estimate of the partition width would be its expected value $E_w = \left(\frac{w_{min}+w_{max}}{2}\right)$. Similarly, the height is uniform in the range $[h_{min}, h_{max}]$ and its expected value is $E_h = \left(\frac{h_{min}+h_{max}}{2}\right)$. The attacker can calculate the image statistics using uniform size partitions of the size $E_w \times E_h$ to obtain an estimate for the hash values. In Figure 7.13(a), we plot the actual hash values, our estimates and the corresponding difference (i.e. the estimation error). Here, the estimates are obtained by computing the statistics from the closest uniform spaced partition. We note that the error has a much lower dynamic range than the actual value even though the location and size of the estimated partitions are not exactly the same as those used in hash generation. The amount of randomness in the hash values can be characterized by the degree of uncertainty in our estimation. Therefore, the differential entropy of the first aspect of security, $h^{(1)}$, can be numerically obtained by first finding the pdf of the estimation error and then computing the entropy from the pdf For the Lena image, $h^{(1)}$ can be numerically computed to be around 5.74. We also note that $h^{(1)}$ only characterizes one aspect of randomness in the hash values. Therefore, the actual differential entropy of the hash values $\aleph(h_k)$ would be greater than $h^{(1)}$.

The second aspect of the hash security that we consider here is the randomness associated with the order in which the individual hash values are concatenated



(a) The plot of the actual and the estimated image (b) The entropy obtained by modeling the syn-

statistics vector in the first stage of the hashing chronization errors plotted for different parameter scheme along with their differences; for the Lena values of w_{max} and w_{min} with $W = H = 512$, image with $w_{min} = 10$, $w_{max} = 40$, and $W = 512$. $w_{min} = h_{min}$, and $w_{max} = h_{max}$.

Figure 7.13: Security analysis results for Venkatesan's scheme.

together while creating the hash vector. Here, we compare the true hash vectors generated using the randomized block partitions and the ones estimated using uniform partitions and assume that both these hash vectors are obtained using a raster-scan order of the partitioning blocks. It is to be noted that any further permutation of the hash can be factored into the post-processing stage which we shall not consider here as indicated before. A good uniform partition that emulates the randomized partition can be obtained as follows. We model the two-dimensional randomized partitioning as a combination of first partitioning the input image along the vertical direction into rows and then further partitioning each row into blocks. Let M denote the number of rows and N_i denote the number of partitions in the i^{th} row. We can show that the expected value of M and N_i are

$$E(M) = \frac{2H}{h_{min} + h_{max}}, \quad E(N_i) = E(N) = \frac{2W}{w_{min} + w_{max}} \quad \forall\, 1 \leq i \leq M \qquad (7.21)$$

The derivation is presented in Appendix II of this chapter.

Since, we use a uniform partition to approximate the randomized partition, there will be synchronization errors in each row of the estimated partition. Let us now denote the amount of synchronization errors in the n^{th} row by Y_n. The synchronization error is cumulative and can be written as

$$Y_n = \sum_{i=1}^{n} (N_i - m_N). \tag{7.22}$$

In order to facilitate combining the security analysis of the synchronization error with the differential entropy $h^{(1)}$ derived for first security aspect, we provide a continuous approximation of Y_n and bound its maximum amount of uncertainty. We note that among all continuous random variables with the same variance, the Gaussian distribution has the maximum differential entropy; and that the differential entropy is completely specified by the determinant of its correlation matrix. So we construct a $M \times M$ correlation matrix R_Y for the set of random variables $\{Y_1, Y_2, \ldots, Y_M\}$,

$$R_Y(i,j) = E(Y_i Y_j) = \min(i,j)\sigma_N^2. \tag{7.23}$$

Here, σ_N^2 denotes the variance of N_i and can be computed from its probability mass function (pmf) given in (7.38) of Appendix II of this chapter. It can be shown that $|R_Y| = \sigma_N^{2M}$. Therefore, using the Gaussian upper bound, the differential entropy of the stage ($h^{(2)}$) considering the synchronization errors alone is given by

$$h^{(2)} \leq \frac{1}{2}\log_2(2\pi e\sigma_N^2) + \frac{1}{2m_M}\log_2\left(1 + \frac{1}{12\sigma_N^2}\right). \tag{7.24}$$

In Figure 7.13(b), we show the plot of the upper bound as given by the RHS of (7.24) for different values of w_{min} and w_{max}. We observe that the upper bound heavily depends on the value of the variance σ_N^2. For very small w_{max}, we have

Table 7.4: Comparison of differential entropy of various hashing schemes shown for three different images.

Hashing algorithm	Differential entropy		
	Lena	Baboon	Peppers
Proposed scheme−1	$8.2 - 15.6$	$13.58 - 16.18$	$8.76 - 15.46$
Proposed scheme−2	16.28	16.39	16.18
Fridrich's scheme [42]	8.31	8.32	8.14
Venkatesan's scheme [138]	$5.74 - 11.48$	$5.96 - 11.70$	$5.65 - 11.39$
Mihçak's algorithm B [98]	8	8	8

$\sigma_N^2 \to 0$ and therefore $h^{(2)} \to -\infty$, suggesting that the hashing algorithm becomes insecure for low σ_N^2. This result is expected because when $w_{max} \approx w_{min}$, the window widths and locations become approximately deterministic and the errors caused by synchronization are small.

Overall, when an attacker replaces the random partitions by uniformly spaced partitions to estimate the hash values, the two aspects of security will both contribute to the uncertainty of the hash algorithm. Thus, the final differential entropy can be approximated by $(h^{(1)} + h^{(2)})$.

The above analysis method can be generalized and extended to other hashing schemes alike. For example, analysis can be applied to the hashing scheme by Mihçak *et al.* [98], which also introduces security by the choice of random regions in the image.

7.3.4 Comparison Results

In this subsection, we compare the security of image hashing schemes in terms of the differential entropy as a metric. We compute the differential entropy of the hash values on the Lena image for various schemes and present the results in Table 7.4.

The differential entropy of the proposed scheme−1 lies in the range $8.2 - 15.6$. This is due to the fact that each hash value in the scheme−1 has different amount of randomness based on the radius on which the summation in (7.5) is performed. If the corresponding Fourier transform coefficients have a higher magnitude, then the variance of the hash values would be larger. Thus some of the hash values can be estimated easily, while it might be difficult to estimate some others. This can be considered as one of the disadvantages of the proposed scheme−1. The disadvantage is overcome in the proposed scheme−2 because the summation is done over randomly chosen subsets and thus all the hash values would have a similar amount of randomness. We note that the differential entropy of the feature extraction stage of the proposed scheme−2 is higher than that of the scheme−1. This is expected because in the proposed scheme−2, the random weights are scaled by larger factors and thus the overall variance of the hash values would be higher

Next, we observe that the differential entropy of the proposed scheme−2 is greater than that of Fridrich's scheme. This can be attributed to the low-pass filtering operations in Fridrich's scheme that reduces the variance of the random variables and hence its entropy. The differential entropy of Venkatesan's scheme is lower than those of proposed schemes. This is because, even without the knowledge of the exact block partitions, the image statistics in Venkatesan's scheme can be estimated to reasonable accuracy. On the other hand, in the proposed schemes,

the attackers need to guess the random variables in computing features (such as β_{ik}).

Notice that we only consider the security of the feature extraction stage in this work. It should be noted that while random permutation or other techniques alike can be applied to any scheme to bring further randomness, such post-processing does not change the relative security results obtained in this work. If the type of quantization and/or quantization step size employed by various schemes are not identical, the gap between the security metric for these schemes may change and can be further analyzed.

7.4 Discussions

7.4.1 Trade-off Between Robustness and Security

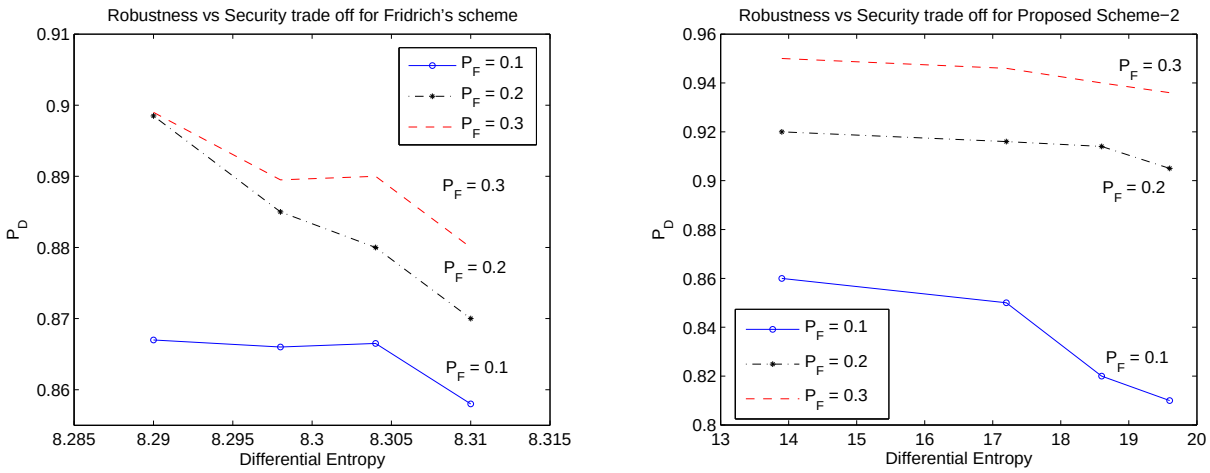
In this section, we jointly consider the two main performance criteria for image hashing, namely, robustness and security. We observe a trade-off between the two criteria for each hashing scheme and illustrate this phenomenon with some examples.

In Figure 7.14(a), we show the trade-off between robustness and security for the Fridrich's scheme [42]. The scheme was simulated for different orders of averaging filter; and the ROC and the differential entropy was obtained in each case. The ROC was sampled to obtain the probabilities of correct decisions P_D for three different probabilities of false alarm P_F, and plotted with respect to the differential entropy. We observe that as the robustness increases, the scheme becomes less secure and vice-versa. This trend is expected because on increasing the order of the averaging filters, the patterns $Y^{(r)}$ become more smooth making the scheme more

robust to content-preserving manipulations like the ones in Table 7.1. However, the scheme becomes less secure because the smooth patterns $Y^{(r)}$ would be less random.

Similar behavior can also be observed for the proposed scheme-2. The performance of the scheme was studied for different parameter values; and the ROC and the differential entropy were obtained in each case. As shown in the Figure 7.14(b), we observe that for a fixed P_F, as we increase the variance of the random weights β_{ik}, the differential entropy increases and the robustness decreases. However, it is to be noted that proposed scheme exhibits a better trade-off compared to Fridrich's scheme. This is evident by comparing the X-axis of Figure 7.14(a) and (b). We observe that proposed scheme-2 is more secure than the Fridrich's scheme for the same amount of robustness. This demonstrates the advantages of the proposed scheme.

The robustness results in Figure 7.10 and the differential entropy values in Table 7.4 show that the proposed scheme-2 provides better tradeoff between robustness and security against guessing than the proposed scheme-1. This is attributed to the fact that the circular summation along the θ-axis in proposed scheme-2 can generate more robust features. In the mean time, we also remark that the circular summation is a double-edged sword and may reduce the resilience against collision and forgery attacks. It is possible for malicious attackers to perform meaningful changes by altering individual values of the Fourier transform coefficients while preserving the overall sum. In contrast, the proposed scheme-1 is more resilient to such collision attacks, as the weights of the summation are random and depend on a secret key unknown to adversaries. A possible improvement is to employ a weighted circular summation with gradually changing weights, where the varying



(a) (b)

Figure 7.14: Robustness and security trade-off for (a) Fridrich's scheme (b) Proposed scheme−2.

trend of the weights is specified by a secret key. This hybrid scheme can combine the advantages of the two proposed schemes, improving the collision resistance compared to scheme−2 and also the robustness compared to scheme−1.

7.4.2 Extending the Security Analysis to Quantization Algorithms

We have shown that the differential entropy can be used as a metric to study the security of the feature extraction stage in image hashing. In this section, we extend the security analysis beyond the feature extraction stage and show that entropy can be used as a metric to study the degree of security of the quantization stage that follows feature extraction.

As an example, we consider the randomized quantization algorithm proposed in [97], which is an adaptive quantization algorithm that takes into account the

distribution of the input data. The quantization bins $[\Delta_{i-1}, \Delta_i]$ are designed so that $\int_{\Delta_{i-1}}^{\Delta_i} p_X(x)dx = \frac{1}{Q}$, where Q is the number of quantization levels and $p_X(\cdot)$ is the pdf of the input data X. The central points $\{C_i\}$ are defined so as to make $\int_{\Delta_{i-1}}^{C_i} p_X(x)dx = \int_{C_i}^{\Delta_i} p_X(x)dx = \frac{1}{2Q}$; and the randomization interval $[A_i, B_i]$ are chosen such that $\int_{A_i}^{\Delta_i} p_X(x)dx = \int_{\Delta_i}^{B_i} p_X(x)dx = \frac{r}{Q}$, where $r \leq \frac{1}{2}$ is a randomization parameter. The overall quantization method can be expressed as

$$q(x) = \begin{cases} i-1 & \text{w.p.} \quad 1 & \text{if } C_i \leq x \leq A_i, \\ i-1 & \text{w.p.} \quad \left(\frac{Q}{2r}\int_x^{B_i} p_X(t)dt\right) & \text{if } A_i \leq x \leq B_i, \\ i & \text{w.p.} \quad \left(\frac{Q}{2r}\int_{A_i}^{x} p_X(t)dt\right) & \text{if } A_i \leq x \leq B_i, \\ i & \text{w.p.} \quad 1 & \text{if } B_i \leq x \leq C_{i+1}. \end{cases} \tag{7.25}$$

We again use the conditional entropy $\aleph(h_k|I)$ as a security metric. Based on the detailed derivation in Appendix III of this chapter, we can show that

$$H(q(X)|X) = r\log_2(e), \tag{7.26}$$

which quantifies the amount of randomness introduced by the randomized quantization. We note that the conditional entropy is directly proportional on the randomization parameter r, and is independent of the source distribution. Other quantization algorithms can be analyzed similarly using conditional entropy as a metric.

7.4.3 Further Discussions on Hash Security

In this work, we have considered the conditional entropy of the hash values as a metric to study security. Our analysis is based on the premise that the adversary knows the image and the hashing algorithm being used and does not know the key

used in generating the hash. Therefore, in our analysis, the adversary does not have access to the actual hash values and tries to estimate them based on his knowledge. Alternatively, we can evaluate the security of a hashing scheme by measuring the conditional entropy of the *hashing key* when the image, the hashing algorithm and output hash values are known. This conditional entropy can be written as $\aleph(K|(I,h))$, where K denotes the key, I the image, and h the corresponding hash value. In reality, if more information is available to the adversary, he/she may be able to come up with more sophisticated attacks to break the hashing algorithm. In such a case, the conditional entropy of the key will reduce with the increase in the number of observed image/hash pairs. Thus, $\aleph(K|(I_1,h_1),(I_2,h_2),...(I_n,h_n))$ is a monotonically decreasing function with n. When n is large enough, it would be possible to uniquely identify the key K with very high probability. This is analogous to Shannon's discussion on secrecy system and his definition of unicity distance [118]. Along these lines, we may define another notion of hashing security by requiring that the conditional entropy $\aleph(K|(I_1,h_1),(I_2,h_2),...(I_n,h_n))$ is not negligible as long as the number of observed image/hash pairs, n, is upper bounded by a polynomial in key length. We note that for image hashing and other types of multimedia hashing, an adversary may not need to exactly recover the key in order to estimate a hash. The estimation type of attack introduced in [116] is clearly an example.

7.5 Chapter Summary

Robustness and security are two important requirements for image hashing algorithms in applications involving authentication, watermarking, and image databases. In this chapter, we have developed a new image hashing schemes that has improved

robustness and security features. We show that the proposed schemes is resilient to moderate filtering, and compression operations, and common geometric operations up to 10 degrees of rotation and 20 percent of cropping. The proposed hashing scheme also has good discriminative capabilities and can identify malicious manipulations, such as cut-and-paste type of editing, that do not preserve the content of the image. In addition to the study on robustness, we have introduced a general framework for analyzing the security in image hashing. We derive analytical expressions using differential entropy as a metric to study the security of the feature extraction stage for both the proposed schemes and several existing representative schemes. Our studies have shown that the proposed image hashing algorithm is highly secure in terms of this metric. The analysis can also be extended to incorporate other stages of the hashing operation, such as randomized quantization.

Overall, we developed a new image hashing algorithm. It is more robust compared to existing image hashing schemes, and at the same time, it is also secure against estimation and forgery attacks. Thus, it can provide a robust and secure representation of images for numerous applications.

Appendix: Details on Modeling and Derivations

Appendix I: Deriving the Security Metric for the Fridrich's scheme [42]

In Fridrich's scheme, key-dependent pseudo-random patterns $X^{(r)}(r = 1, 2, \ldots N)$ of the same size of the input image are first generated. These pseudo-random

patterns have uniform distributed pixel values. These patterns are then spatially averaged with a $m \times n$ low-pass filter $\{\alpha_{ij}\}$ to obtain zero-mean random images $[Y^{(r)}]_{kl}$

$$Y_{kl}^{(r)} = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} \alpha_{ij} X_{i+k,j+l}^{(r)}. \tag{7.27}$$

The input image I is projected on the N smooth patterns $\{Y^{(r)}\}$ to obtain the intermediate hash values h_r as given by

$$h_r = \sum_{k=1}^{H} \sum_{l=1}^{W} Y_{kl}^{(r)} I_{kl}. \tag{7.28}$$

These intermediate hash values are then quantized to generate the final hash. In our analysis, we model the intermediate hash values h_r as random variables and find its differential entropy to generate the security metric. The hash values h_r in (7.28) can be rewritten as

$$h_r = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} \alpha_{ij} V_{ij}^{(r)}, \tag{7.29}$$

where the random variables $V_{ij}^{(r)}$ are defined as

$$V_{ij}^{(r)} = \sum_{k=1}^{H} \sum_{l=1}^{W} X_{i+k,j+l}^{(r)} I_{kl}. \tag{7.30}$$

We observe that $V_{ij}^{(r)}$ is a weighted sum of $W \times H$ uniformly distributed random variables $\{X_{ij}^{(r)}\}$ with the weights determined by the image pixel values (I_{kl}). According to the Central Limit Theorem, we approximate $V_{ij}^{(r)}$ to be Gaussian distributed, with mean $m_{ij}^{(r)}$ and variance $\sigma_{ij}^{2(r)}$ that can be shown to be

$$\begin{aligned} m_{ij}^{(r)} &= E(V_{ij}^{(r)}) = \frac{1}{2} \left(\sum_{k=1}^{H} \sum_{l=1}^{W} I_{kl} \right), \\ \sigma_{ij}^{2(r)} &= \frac{1}{12} \left(\sum_{k=1}^{H} \sum_{l=1}^{W} I_{kl}^2 \right). \end{aligned} \tag{7.31}$$

We also note that all $\{V_{ij}^{(r)}\}$ are identically distributed, but are not independent since the same random variables $\{X_{ij}^{(r)}\}$ are used to generate various $V_{ij}^{(r)}$. The dependence among the variables $\{V_{ij}^{(r)}\}$ can be expressed in terms of their correlation given by

$$E(V_{ij}^{(r)}V_{ab}^{(r)}) = \frac{1}{12}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}I_{i+k-a,j+l-b} + \left(\frac{1}{2}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}\right)^2. \tag{7.32}$$

Now, from (7.29), we see that h_r is a weighted sum of $m \times n$ Gaussian distributed random variables. So h_r is also Gaussian and its differential entropy is completely specified by its variance. The variance of h_r can be computed as

$$\begin{aligned}
\sigma_{h_r}^2 &= E(h_r^2) - m_{h_r}^2 \\
&= E\left(\sum_{i=-\lfloor\frac{m}{2}\rfloor}^{\lfloor\frac{m}{2}\rfloor}\sum_{j=-\lfloor\frac{n}{2}\rfloor}^{\lfloor\frac{n}{2}\rfloor}\alpha_{ij}V_{ij}^{(r)}\right)^2 - \left(\frac{1}{2}\sum_{k=1}^{H}\sum_{l=1}^{W} I_{kl}\right)^2 \\
&= \frac{1}{12}\sum_{p=1}^{H}\sum_{q=1}^{W} I_{pq}I_{pq}^{(\alpha\alpha)}, \quad \text{where} \qquad (7.33)\\
I_{pq}^{(\alpha\alpha)} &= \sum_{i,k=-\lfloor\frac{m}{2}\rfloor}^{\lfloor\frac{m}{2}\rfloor}\sum_{j,l=-\lfloor\frac{n}{2}\rfloor}^{\lfloor\frac{n}{2}\rfloor}\alpha_{ij}\alpha_{kl}I_{i+p-k,j+q-l}. \qquad (7.34)
\end{aligned}$$

Note that $I^{(\alpha\alpha)}$ is the image obtained by filtering I the image twice with the filter $\{\alpha_{ij}\}$. Using the result in (7.33), we obtain the differential entropy of h_r as

$$\aleph(h_r) \approx \frac{1}{2}\log_2\left(2\pi e\,\frac{1}{12}\sum_{p=1}^{H}\sum_{q=1}^{W} I_{pq}I_{pq}^{(\alpha\alpha)}\right). \tag{7.35}$$

Appendix II: Model for Block partitioning in Venkatesan's scheme [138]

As indicated in Section 7.3.3, we approximate the 2-D block partitioning as a combination of two 1-D problems, namely, partitioning along the horizontal direction

and then along the vertical direction. To model the partition along the width of the image, we divide the space $(0, W)$ into several regions by successively generating random numbers $\{U_k\}$ as shown in Figure 7.15, uniformly distributed in $[w_{min}, w_{max}]$, and w_{min} and w_{max} are the minimum and the maximum widths of the random blocks. The location of the n^{th} partition is then given by a set of random variables T_n, where $T_n = \sum_{k=1}^{n} U_k$. Since T_n is the sum of n uniformly distributed random variables, we approximate T_n with a Gaussian distribution. Its mean m_{T_n} and variance $\sigma^2_{T_n}$ can be shown to be

$$m_{T_n} = \frac{n}{2}(w_{min} + w_{max}), \quad \sigma^2_{T_n} = \frac{n}{12}(w_{max} - w_{min})^2. \tag{7.36}$$

Let N_i denote the number of partitions in the i^{th} row. Using the distribution of T_n and noting that N_i is also the index for the last partition in the row, we can write the pmf of N_i as

$$\begin{aligned} P(N_i = n) &= Pr\left(T_n < W < T_{n+1}\right) = Pr\left(\max(W - T_n, w_{min}) < U_{n+1} < w_{max}\right) \\ &= \int_{W-w_{max}}^{W-w_{min}} P(W - t < U_{n+1} < w_{max}) f_{T_n}(t)\ dt \\ &+ \int_{W-w_{min}}^{W} P(w_{min} < U_{n+1} < w_{max}) f_{T_n}(t)\ dt, \end{aligned} \tag{7.37}$$

where $f_{T_n}(\cdot)$ is the pdf of T_n. Using the Gaussian assumption on T_n, the above expression can be simplified as

$$\begin{aligned} P(N_i = n) &= \frac{\sigma_n}{\sqrt{2\pi}(w_{max} - w_{min})} \exp\left(-\frac{(W - w_{max} - m_{T_n})^2}{2\sigma^2_{T_n}}\right) \\ &- \frac{\sigma_n}{\sqrt{2\pi}(w_{max} - w_{min})} \exp\left(-\frac{(W - w_{min} - m_{T_n})^2}{2\sigma^2_{T_n}}\right) \\ &+ \frac{w_{max} + m_{T_n} - W}{w_{max} - w_{min}} \left(F_{T_n}(W - w_{min}) - F_{T_n}(W - w_{max})\right) \\ &+ \left(F_{T_n}(W) - F_{T_n}(W - w_{min})\right), \end{aligned} \tag{7.38}$$

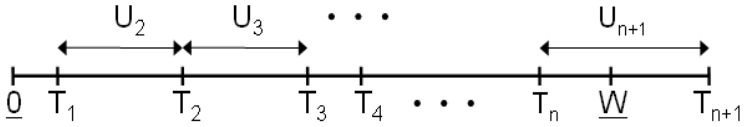


Figure 7.15: Simplified model of the block partitioning algorithm in Venkatesan's scheme [138]

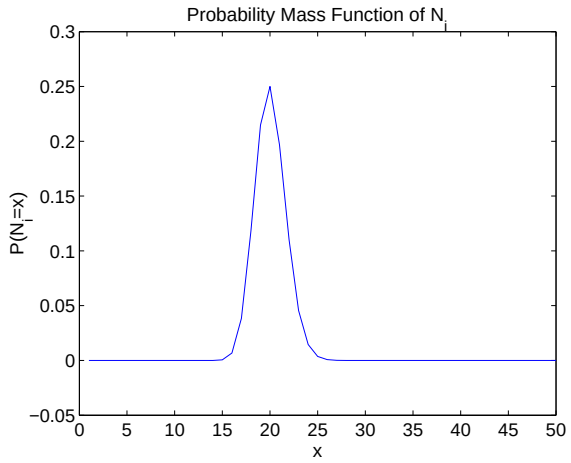


Figure 7.16: The plot of the pmf of N_i—the number of blocks in i^{th} row, where the parameters are $w_{min} = 10$, $w_{max} = 40$, and $W = 512$. Note that the random variable N_i has a very small variance and hence the mean would be a good estimate.

where $F_{T_n}(x)$ is the cumulative distribution function (cdf) of T_n, and is given by

$$F_{T_n}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\left(\frac{x-m_{T_n}}{\sigma_{T_n}}\right)} \exp\left(-\frac{z^2}{2}\right) dz. \tag{7.39}$$

The plot of the pmf of N_i is shown in Figure 7.16. From this pmf, we can derive the expected value of N_i as $E(N_i) = \frac{2W}{w_{min}+w_{max}}$.

Appendix III: Deriving the Security Metric for Randomized Quantization [97]

In this appendix, we provide the detailed derivations of the conditional entropy for the randomized quantization algorithm [97]. The conditional entropy $H(q(X)|X)$ can be written as

$$\begin{aligned} H(q(X)|X) &= \int_{x\in\Re} H(q(X)|X=x)p_X(x)dx \\ &= \sum_{i=1}^{Q}\int_{C_i}^{C_{i+1}} H(q(X)|X=x)p_X(x)dx \\ &= \sum_{i=1}^{Q}\int_{A_i}^{B_i} H(q(X)|X=x)p_X(x)dx, \end{aligned} \tag{7.40}$$

where $p_X(\cdot)$ denotes the pdf of the input data X. The last step follows from (7.25) since the quantizer $q(X)$ is random only in the interval $A_i \leq x \leq B_i$. Now, we note that in this interval, $q(X)$ takes a value i with probability $p_i = (P_X(x) - P_X(A_i))\frac{Q}{2r}$, and a value $(i-1)$ with probability $(1-p_i)$. Therefore, (7.40) can be calculated and simplified as

$$\begin{aligned} H(q(X)|X) &= -\sum_{i=1}^{Q}\int_{A_i}^{B_i} (p_i\log_2(p_i) + (1-p_i)\log_2(1-p_i))p_X(x)dx \\ &= r\log_2(e). \end{aligned} \tag{7.41}$$

Chapter 8

Conclusions and Future Perspectives

In this dissertation, we have introduced two new frameworks for forensic analysis of digital camera images based on intrinsic and extrinsic fingerprints.

We consider the problem of component forensics and propose a set of forensic signal processing techniques based on intrinsic fingerprinting to identify the algorithms and parameters employed in the individual processing modules of digital devices. We particularly focus on digital cameras for this dissertation and propose a non-intrusive methodology to estimate the parameters of camera's color filter array and color interpolation modules; these parameters form the intrinsic fingerprint traces of the digital camera. We show through detailed simulations with 19 camera models of nine different brands that the proposed algorithms can authenticate the source camera and identify the exact brand with 90% accuracy. Our analysis also suggests that there is a considerable degree of similarity within the cameras of the same brand and some level of resemblance among cameras from different manufacturers.

Building upon component forensics, we introduce a new formulation to study the problem of image authenticity. The proposed formulation is based on the observation that each in-camera and post-camera processing operation leave some distinct intrinsic fingerprint traces on the final image. Using appropriate models, we present techniques to estimate the in-camera component parameters and the linear shift-invariant approximation of the post-camera manipulations. We show that evidence obtained from such forensic analysis is used to build a forensic testbed to identify the image acquisition source (whether the image was captured using a camera, cell phone camera, scanner, or generated via computer graphics?), the brand and model of the imaging device, and to determine if there has been any post-device processing such as tampering or steganographic embedding. Overall, our proposed techniques provides a common framework for a broad range of forensic analysis on digital images.

We then present a generalized theoretical analysis to gain a concrete understanding about component forensics and to answer a number of fundamental questions related to what processing operations can and cannot be identified and under what conditions. We define formal notions of classifiability of components and present bounds on parameter estimation accuracies. Developing upon notions from the theoretical analysis, we present techniques for robustly estimating the component parameters via semi non-intrusive forensics. We believe that such component forensic analysis would provide a great source of information for patent infringement cases, intellectual property rights management, and technology evolution studies for digital media and push the frontiers of multimedia forensics to gain a deeper understanding of information processing chain.

While the presented component forensics and intrinsic fingerprinting techniques

can be employed to determine the source and the authenticity of images just based on the output data, their accuracies are limited by theoretical performance bounds. Extrinsic fingerprinting helps bridge the performance gap by employing external signals, added to the image after capture, to establish the authenticity of the image. In this dissertation, we design a new content-based image authentication scheme based on image hashing and show that the proposed scheme is collision-resistant, robust to common signal processing operations, and secure against estimation and forgery attacks. Combined with intrinsic fingerprint techniques, extrinsic fingerprinting provides a universal framework for digital image forensics for a wide range of applications.

The main contributions of the thesis are as follows:

- Introduced component forensics as a new methodology for multimedia forensics, aiming at identifying algorithms and parameters in each component of an information processing chain.

- Proposed algorithms to non-intrusively estimate the parameters of in-camera components such as the color filter array and the color interpolation based solely on the output data.

- Applied the estimated in-camera parameters for several forensic tasks, including camera identification and technology infringement/licensing forensics, and to design a universal framework for image acquisition forensics.

- Introduced methods to detect post-camera processing operations by modeling them as a linear shift invariant system and casting the problem into a blind deconvolution framework; and showed that the estimated manipulation filter coefficients can efficiently differentiate between processed images and direct

camera outputs.

- Developed a new theoretical framework for multimedia forensics based on estimation and pattern classification theories. This is the first work in literature to look into theoretical analysis of multimedia forensics.

- Introduced the concept of semi non-intrusive forensics and devised methods to design optimal inputs for semi non-intrusive forensics.

- Presented a new robust and secure hash as an extrinsic fingerprint and showed that the proposed hash is resilient to geometric and filtering operations in images.

- Introduced a systematic evaluation of the security of image hash functions and demonstrated the trade-offs between robustness and security in several hashing schemes.

Based on the study of this dissertation, there are several aspects of multimedia forensics that can be further explored. In our work, we have mainly focussed on digital cameras. However, the fundamental principles of intrinsic fingerprinting and component forensics can be widely applicable to a range of other imaging devices such as scanners, cell phone cameras, and video recorders; and display devices including projectors and Liquid Crystal Display (LCD) screens. In our recent work, we have extended the forensic methodology beyond cameras and employed it for cell phone cameras [94] and with image scanners [54, 55] with very encouraging results. A promising next step is to go beyond still images and apply the analysis to digital video data. Video brings in several additional challenges due to its time domain features. Therefore, a more sophisticated imaging model incorporating the effects of time domain would be necessary to perform forensic analysis of video. The time

domain also allows for better attacks and more possibilities for the attackers, and it would be interesting to design and introduce methods for component forensics of digital video that would be robust to improved and more targeted attacks.

The research on component forensics and intrinsic fingerprinting presented in this thesis can also be applicable to a number of interesting problems from communications and networking to biology and web design. For instance, transmitting data from the sender to the receiver involves a series of processing operations that include source coding, channel coding, message modulation onto a carrier signal, physical transmission over a channel (wireline or wireless), demodulation, and decoding. Forensic analysis on the various components of the information processing chain, to estimate the parameters such components as source coding, channel coding, the message modulation scheme, and the channel parameters, just based on the received signal can help identify the nature of the source and further help establish the integrity of the message. The proposed theoretical framework can also be extended to other applications such as to analyze biological processes.

In this thesis, we have examined both intrinsic and extrinsic fingerprint approaches for multimedia forensics and demonstrated the applicability, advantages, and drawbacks of these frameworks. A natural extension of this work is to examine a joint intrinsic-extrinsic framework for forensic analysis that can combine the advantages of the two frameworks. One step in this direction is to design *forensic hashes*. Just as the image hash is a content-based compact representation of an image with applications in image authentication, the forensic hash is a short representation of the data focussed on gaining a better understanding of the information processing chain to answer forensic questions regarding how an image was generated; from where an image was from; what has been done on the image since

its creation, by whom, when and how. The forensic hash can be designed to be an intrinsic device-specific fingerprint or an extrinsic fingerprint that is added to the image at the time of capture. This new hash can then be employed to identify the tell-tale clues about the various processing operations that the image/video has gone through. It would be interesting to examine the design and performance of these joint fingerprints for various forensic tasks.

BIBLIOGRAPHY

[1] General Information Concerning Patents. Brochure available online at the U.S. patents website: `http://www.uspto.gov/web/offices/pac/doc/general/infringe.htm`.

[2] Information about Foveon X3 Sensors. Available online at `http://www.foveon.com/`.

[3] Information about Fujifilm Super CCD Cameras. Available online at `http://www.fujifilm.com/superccd/`.

[4] Business Week News: "How Ampex squeezes out cash: It's suing high-tech giants that rely on its patents. Will its stock keep on soaring?" `http://www.businessweek.com`, last accessed, April 2005.

[5] CNET News Article: "Who will become the Intel of Photography?", `http://news.cnet.com/picture+this+a+new+breed+of+cameras/2009-1006_3-5559249.html`, last accessed, February 2005.

[6] J. Adams, K. Parulski, and K. Spaulding. Color Processing in Digital Cameras. *IEEE Micro*, 18(6):20–30, November-December 1998.

[7] J. E. Adams. Interaction Between Color Plane Interpolation and Other Image Processing Functions in Electronic Photography. In *Proceedings of the SPIE, Cameras and Systems for Electronic Photography and Scientific Imaging*, volume 2416, pages 144–151, San Jose, CA, February 1995.

[8] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur. A Classifier Design for Detecting Image Manipulations. In *IEEE International Conference on Image Processing (ICIP)*, volume 4, pages 2645–2648, Singapore, Singapore, October 2004.

[9] I. Avcibas, N. Memon, and B. Sankur. Steganalysis using Image Quality Metrics. *IEEE Trans. on Image Processing*, 12(2):221–229, February 2003.

[10] I. Avcibas, B. Sankur, and K. Sayood. Statistical Evaluation of Image Quality Metrics. *Journal of Electronic Imaging*, 11(2):206–223, April 2002.

[11] G. R. Ayers and J. C. Dainty. Iterative Blind Deconvolution Method and its Applications. *Optics Letters*, 13(7):547–549, July 1988.

[12] K. Barnard. Computational Colour Constancy: Taking Theory into Practice. *MSc thesis, Simon Fraser University, School of Computing*, 1995.

[13] B. E. Bayer. Color Imaging Array. In *U.S. Patent no. 3,971,065*, July 1976.

[14] S. Bayram, H. T. Sencar, and N. Memon. Improvements on Source Camera-model Identification based on CFA Interpolation. In *Proceedings of the WG 11.9 Intl. Conference on Digital Forensics*, Orlando, FL, January 2006.

[15] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas. Source Camera Identification based on CFA Interpolation. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 69–72, Genoa, Italy, September 2005.

[16] S. Bhattacharjee and M. Kutter. Compression Tolerant Image Authentication. In *IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 435–439, Chicago, IL, October 1998.

[17] R. E. Blahut. *Theory and Practice of Error-Control Codes*. Addison-Wesley, 1983.

[18] D. H. Brainard and B. A. Wandell. Analysis of the Retinex Theory of Color Vision. *Journal of the Optical Society of America*, 3(10):1651–1661, October 1986.

[19] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998.

[20] J. Cannons and P. Moulin. Design and Statistical Analysis of a Hash-Aided Image Watermarking System. *IEEE Trans. on Image Processing*, 13(10):1393–1408, October 2004.

[21] O. Celiktutan, I. Avcibas, B. Sankur, N. P. Ayerden, and C. Capar. Source Cell-phone Identification. In *Proceedings of the IEEE Conference on Signal Processing and Communications Applications*, pages 1–3, Antalya, Turkey, April 2006.

[22] C-C. Chang and C-J. Lin. LIBSVM: A Library for Support Vector Machines, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[23] E. Chang, S. Cheung, and D. Y. Pan. Color Filter Array Recovery using a Threshold-based Variable Number of Gradients. In *Proceedings of the SPIE, Sensors, Cameras, and Applications for Digital Photography*, volume 3650, pages 36–43, San Jose, CA, March 1999.

[24] B. Chen and G. W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Trans. on Information Theory*, 47(4):1423–1443, May 2001.

[25] S-H. Chen and C-T. Hsu. Source Camera Identification based on Camera Gain Histogram. In *IEEE International Conference on Image Processing (ICIP)*, volume 4, pages 429–432, San Antonio, TX, September 2007.

[26] K. S. Choi, E. Y. Lam, and K. K. Y. Wong. Source Camera Identification using Footprints from Lens Aberration. In *Proceedings of the SPIE, Conference on Digital Photography*, volume 6069, pages 172–179, San Jose, CA, January 2006.

[27] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[28] I. Cox, J. Bloom, and M. Miller. *Digital Watermarking: Principles and Practice*. Morgan Kaufmann, 2002.

[29] I. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, December 1997.

[30] I. J. Cox and J-P. M. G. Linnartz. Public Watermarks and Resistance to Tampering. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 3–6, Washington, DC, October 1997.

[31] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2000.

[32] M. D. Fairchild. *Color Appearance Models*. Addison-Wesley, 1997.

[33] Z. Fan and R. L. de Queiroz. Identification of Bitmap Compression History: JPEG Detection and Quantizer Estimation. *IEEE Trans. on Image Processing*, 12(2):230–235, February 2003.

[34] H. Farid. Blind Inverse Gamma Correction. *IEEE Trans. on Image Processing*, 10(10):1428–1433, October 2001.

[35] H. Farid. Digital Image Ballistics from JPEG Quantization. In *Technical report TR2006-583, Department of Computer Science, Dartmouth College*, 2006.

[36] H. Farid and S. Lyu. Higher-order Wavelet Statistics and their Application to Digital Forensics. In *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI, June 2003.

[37] H. Farid and A.C. Popescu. Blind Removal of Image Non-Linearities. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 76–81, Vancouver, Canada, July 2001.

[38] A. M. Ferman, A. M. Tekalp, and R. Mehrotra. Robust Color Histogram Descriptors for Video Segment Retrieval and Identification. *IEEE Trans. on Image Processing*, 11(5):497–508, May 2002.

[39] G. D. Finlayson, M. S. Drew, and B. V. Funt. Diagonal Transform Suffice for Color Constancy. In *IEEE Conference on Computer Vision*, pages 164–171, Berlin, Germany, May 1993.

[40] J. Fridrich. Image Watermarking for Tamper Detection. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 404–408, Chicago, IL, October 1998.

[41] J. Fridrich. Visual Hash for Oblivious Watermarking. In *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents II*, volume 3971, pages 286–294, San Jose, CA, January 2000.

[42] J. Fridrich and M. Goljan. Robust Hash Functions for Digital Watermarking. In *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, pages 178–183, Las Vegas, NV, March 2000.

[43] J. Fridrich and M. Goljan. Practical Steganalysis of Digital Images - State of the Art. In *Proceedings of the SPIE Conference on Security and Watermarking of Multimedia Contents*, volume 4675, pages 1–13, San Jose, CA, January 2002.

[44] J. Fridrich and M. Goljan. Digital Image Steganography using Stochastic Modulation. In *Proceedings of the SPIE, Security, and Watermarking of Multimedia Contents*, volume 5020, pages 191–202, Santa Clara, CA, January 2003.

[45] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG Images: Breaking the F5 Algorithm. In *Proceedings of the International Workshop on Information Hiding (IHW)*, volume 2578, pages 310–323, Noordwijkerhout, The Netherlands, October 2002.

[46] J. Fridrich, M. Goljan, and D. Soukal. Perturbed Quantization Steganography with Wet Paper Codes. In *Proceedings of the ACM Multimedia Security Workshop*, pages 4–15, Magdeburg, Germany, September 2004.

[47] J. Fridrich, M. Goljan, and D. Soukal. Perturbed Quantization Steganography. *Multimedia Systems*, 11(2):98–107, December 2005.

[48] B. R. Frieden. *Science from Fisher Information*. Cambridge University Press, 2004.

[49] F. Gasparini and R. Schettini. Color Balancing of Digital Photos using Simple Image Statistics. *Pattern Recognition*, 37(6):1201–1217, June 2004.

[50] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh. Methods for Identification of Images Acquired with Digital Cameras. In *Proceedings of the SPIE, Enabling Technologies for Law Enforcement and Security*, volume 4232, pages 505–512, February 2001.

[51] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, 1992.

[52] A. Giannoula, N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis. Watermark Detection for Noisy Interpolated Images. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 53(5):359–363, May 2006.

[53] H. Gou, A. Swaminathan, and M. Wu. Noise Features for Image Tampering Detection and Steganalysis. In *IEEE International Conference on Image Processing (ICIP)*, volume 6, pages 97–100, San Antonio, TX, September 2007.

[54] H. Gou, A. Swaminathan, and M. Wu. Robust Scanner Identification based on Noise Features. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, page 65050S, San Jose, CA, January 2007.

[55] H. Gou, A. Swaminathan, and M. Wu. Intrinsic Sensor Noise Features for Scanner and Scanned Image Forensics. *IEEE Trans. on Information Forensics and Security*, under review, June 2008.

[56] J. F. Hamilton and J. E. Adams. Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera. In *U.S. Patent no. 5,629,734*, May 1997.

[57] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[58] G. E. Healey and R. Kondepudy. Radiometric CCD Camera Calibration and Noise Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[59] S. Hetzl. Steghide, Software available at `steghide.sourceforge.net`.

[60] S. Hetzl and P. Mutzel. A Graph-Theoretic Approach to Steganography. In *9th IFIP Conference on Communications and Multimedia Security and Springer Verlag Lecture Notes in Computer Science*, volume 3677, pages 119–128, Salzburg, Austria, September 2005.

[61] J. B. Hiriart-Urrutty and C. Lemarecal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, 1996.

[62] M. Holliman, N. Memon, and M. M. Yeung. On the Need for Image Dependent Keys for Watermarking. In *Proceedings of Content Security and Data Hiding in Digital Media*, Newark, NJ, May 1999.

[63] A. K. Jain. *Fundamentals of Digital Image Processing*. Pentice Hall, 1989.

[64] F. Jing, M. Li, H-J. Zhang, and B. Zhang. An Efficient and Effective Region-Based Image Retrieval Framework. *IEEE Trans. on Image Processing*, 13(5):699–709, May 2004.

[65] M. Johnson and K. Ramachandran. Dither-based Secure Image Hashing using Distributed Coding. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 751–754, Barcelona, Spain, September 2003.

[66] M. K. Johnson and H. Farid. Exposing Digital Forgeries by Detecting Inconsistencies in Lighting. In *ACM Multimedia and Security Workshop*, pages 1–9, New York, NY, August 2005.

[67] M. K. Johnson and H. Farid. Exposing Digital Forgeries through Chromatic Aberration. In *Proceedings of the ACM Workshop on Multimedia and Security*, pages 48–55, Geneva, Switzerland, September 2006.

[68] S. Kawamura. Capturing Images with Digital Still Cameras. *IEEE Micro*, 18(6):14–19, November-December 1998.

[69] N. Khanna, A. K. Mikkilineni, A. F. Martone, G. N. Ali, G. T-C. Chiu, J. P. Allebach, and E. J. Delp. A Survey of Forensic Characterization Methods for Physical Devices. In *Proceedings of the Digital Forensic Research Workshop*, volume 3, pages 17–28, Lafayette, IN, September 2006.

[70] M. Kharrazi, H. T. Sencar, and N. Memon. Blind Source Camera Identification. In *IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 709–712, Singapore, Singapore, October 2004.

[71] S. S. Kozat, R. Venkatesan, and M. K. Mihçak. Robust Perceptual Image Hashing via Matrix Invariants. In *IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 3443–3446, Singapore, Singapore, October 2004.

[72] D. Kundur and D. Hatzinakos. Blind Image Deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, May 1996.

[73] D. Kundur and D. Hatzinakos. Blind Image Deconvolution Revisited. *IEEE Signal Processing Magazine*, 13(6):61–63, November 1996.

[74] D. Kundur and D. Hatzinakos. A Novel Blind Deconvolution Scheme for Image Restoration using Recursive Filtering. *IEEE Trans. on Signal Processing*, 46(2):375–390, February 1998.

[75] T. V. Lanh, K-S. Chong, S. Emmanuel, and M. S. Kankanhalli. A Survey on Digital Camera Image Forensic Methods. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 16–19, Beijing, China, July 2007.

[76] C. A. Laroche and M. A. Prescott. Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients. In *U.S. Patent no. 5,373,322*, December 1994.

[77] A. Latham. Jpeg Hide and Seek, Software available at `linux01.gwdg.de/alatham/stego`.

[78] F. Lefbvre, J. Czyz, and B. Macq. A Robust Soft Hash Algorithm for Digital Image Signature. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 495–498, Barcelona, Spain, September 2003.

[79] F. Lefbvre, B. Macq, and J-D. Legat. RASH: RAdon Soft Hash Algorithm. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Toulouse, France, September 2002.

[80] C. Y. Lin and S. F. Chang. A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168, February 2001.

[81] C-Y. Lin, M. Wu, J. A. Bloom, M. L. Miller, I. J. Cox, and Y-M. Lui. Rotation, Scale, and Translation Resilient Public Watermarking for Images. *IEEE Trans. on Image Processing*, 10(5):767–782, May 2001.

[82] S. Lin, M. T. Ozsu, V. Oria, and R. Ng. An Extendible Hash for Multi-precision Similarity Querying of Image Databases. In *Proceedings of Very Large Data Bases (VLDB) Conference*, pages 221–230, Roma, Italy, September 2001.

[83] Y. Long and Y. Huang. Image Based Source Camera Identification using Demosaicking. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 419–424, Victoria, Canada, October 2006.

[84] J. Lukas and J. Fridrich. Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In *Proceedings of the Digital Forensics Research Workshop*, Cleveland, OH, August 2003.

[85] J. Lukas, J. Fridrich, and M. Goljan. Determining Digital Image Origin Using Sensor Imperfections. In *Proceedings of the SPIE, image and video communications and processing*, volume 5685, pages 249–260, San Jose, CA, January 2005.

[86] J. Lukas, J. Fridrich, and M. Goljan. Detecting Digital Image Forgeries using Sensor Pattern Noise. In *Proceedings of the SPIE Conference on Security, Stegonography, and Watermarking of Multimedia Contents*, volume 6072, pages 362–372, San Jose, CA, Feburuary 2006.

[87] S. Lyu and H. Farid. How Realistic is Photorealistic? *IEEE Trans. on Signal Processing*, 53(2):845–850, February 2005.

[88] S. Lyu and H. Farid. Steganalysis using Higher-Order Image Statistics. *IEEE Trans. on Information Forensics and Security*, 1(1):111–119, March 2006.

[89] W. Macy and O. Rashkowskiy. Software Correction of Image Distortion in Digital Cameras. In *U.S. Patent no. 6,538,691*, March 2003.

[90] M. Malkin and R. Venkatesan. The Randlet Transform: Applications to Universal Perceptual Hashing and Image Authentication. In *Proceedings of the Allerton Conference on Communications, Control, and Computing*, pages 367–378, Monticello, IL, September 2004.

[91] L. M. Marvel, C. G. Boncelet Jr., and C. T. Retter. Spread Spectrum Image Steganography. *IEEE Trans. on Image Processing*, 8(8):1075–1083, August 1999.

[92] T. A. Matraszek, D. R. Cok, and R. T. Gray. Gradient Based Method for Providing Values for Unknown Pixels in a Digital Image. In *U.S. Patent no. 5,875,040*, February 1999.

[93] D. F. McGahn. Copyright Infringement of Protected Computer Software: An Analytical Method to Determine Substantial Similarity. *Rutgers Computer and Technology Law Journal*, 21(1):88–142, 1995.

[94] C. E. McKay, A. Swaminathan, H. Gou, and M. Wu. Image Acquisition Forensics: Forensic Analysis to Identify Imaging Source. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1657–1660, Las Vegas, NV, March 2008.

[95] A. Meixner and A. Uhl. Analysis of a Wavelet Based Robust Hash Algorithm. In *Proceedings of the SPIE Conference on Security, Stegonography, and Watermarking of Multimedia Contents*, volume 5306, pages 772–783, San Jose, CA, January 2004.

[96] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[97] M. K. Mihçak and R. Venkatesan. A Tool for Robust Audio Information Hiding: A Perceptual Audio Hashing Algorithm. In *Proceedings of the Information Hiding Workshop (IHW) and Lecture Notes in Computer Science*, volume 2137, pages 51–65, Pittsburgh, PA, April 2001.

[98] M. K. Mihçak and R. Venkatesan. New Iterative Geometric Methods for Robust Perceptual Image Hashing. In *Proceedings of ACM Workshop on Security and Privacy in Digital Rights Management and Lecture Notes in Computer Science*, volume 2320, pages 13–21, Philadelphia, PA, November 2001.

[99] V. Monga, A. Banerjee, and B. L. Evans. A Clustering Based Approach to Perceptual Image Hashing. *IEEE Trans. on Information Forensics and Security*, 1(1):68–79, March 2006.

[100] V. Monga and B. L. Evans. Robust Perceptual Image Hashing Using Feature Points. In *IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 677–680, Singapore, Singapore, October 2004.

[101] T-T. Ng, S-F. Chang, J. Hsu, and M. Pepeljugoski. Columbia Photographic Images and Photorealistic Computer Graphics Dataset. In *ADVENT Technical Report 205-2004-5*, Columbia University, February 2005.

[102] T-T. Ng, S-F. Chang, J. Hsu, L. Xie, and M-P. Tsui. Physics-motivated Features for Distinguishing Photographic Images and Computer Graphics. In *ACM Multimedia*, pages 239–248, Singapore, Singapore, November 2005.

[103] P. Nillius and J. O. Eklundh. Automatic Estimation of the Projected Light Source Direction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1076–1083, Kauai, HI, December 2001.

[104] R. M. Nosofsky. Tests of an Exemplar Model for Relating Perceptual Classification and Recognition Memory. *Journal of Experimental Psychology: Human Perception and Performance*, 17(1):3–27, 1991.

[105] J. J. K. O'Ruanaidh and T. Pun. Rotation, Scale and Translation Invariant Digital Image Watermarking. In *IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 536–539, Washington, DC, October 1997.

[106] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill Publications, 2002.

[107] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on Copyright Marking Systems. In *Proceedings of the International Workshop on Information Hiding (IHW) and Lecture Notes in Computer Science*, volume 1525, pages 218–238, Portland, OR, April 1998.

[108] J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers (Neural Information Processing): MIT press*, pages 61–74, Cambridge, MA, 1999.

[109] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer Verlag, 1994.

[110] A. C. Popescu and H. Farid. Statistical Tools for Digital Forensics. In *Proceedings of the International Workshop on Digital Watermarking (IWDW) and Lecture notes in Computer Science*, volume 3200, pages 128–147, Toronto, Canada, May 2004.

[111] A. C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Traces of Re-sampling. *IEEE Trans. on Information Forensics and Security*, 53(2):758–767, February 2005.

[112] A. C. Popescu and H. Farid. Exposing Digital Forgeries in Color Filter Array Interpolated Images. *IEEE Trans. on Signal Processing*, 53(10):3948–3959, October 2005.

[113] N. Provos. Outguess, Software available at `www.outguess.org`.

[114] N. Provos and P. Honeyman. Hide and Seek: An Introduction to Steganography. *IEEE Security and Privacy Magazine*, 1(3):32–44, May-June 2003.

[115] M. P. Queluz. Towards Robust, Content Based Techniques for Image Authentication. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 297–302, Redondo Beach, CA, December 1998.

[116] R. Radhakrishnan, Z. Xiong, and N. Memon. On the Security of the Visual Hash Function. In *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents*, volume 5020, pages 644–652, Santa Clara, CA, January 2003.

[117] M. Schneider and S-F. Chang. A Robust Content Based Digital Signature for Image Authentication. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 227–230, Lausanne, Switzerland, September 1996.

[118] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Techical Journal*, 28:656–715, October 1949.

[119] K. Su, D. Kundur, and D. Hatzinakos. Statistical Invisibility for Collusion-resistant Digital Video Watermarking. *IEEE Trans. on Multimedia*, 7(1):43–51, February 2005.

[120] A. Swaminathan, Y. Mao, and M. Wu. Image Hashing Resilient to Geometric and Filtering Operations. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, volume 2, pages 17–20, Siena, Italy, September 2004.

[121] A. Swaminathan, Y. Mao, and M. Wu. Security of Feature Extraction in Image Hashing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1041–1044, Philadelphia, PA, March 2005.

[122] A. Swaminathan, Y. Mao, and M. Wu. Robust and Secure Image Hashing. *IEEE Trans. on Information Forensics and Security*, 1(2):215–230, June 2006.

[123] A. Swaminathan, M. Wu, and K. J. R. Liu. Component Forensics of Digital Cameras: A Non-intrusive Approach. In *Proceedings of the Conference on Information Sciences and Systems (CISS)*, pages 1194–1199, Princeton, NJ, March 2006.

[124] A. Swaminathan, M. Wu, and K. J. R. Liu. Image Tampering Identification using Blind Deconvolution. In *IEEE International Conference on Image Processing (ICIP)*, pages 2309–2312, Atlanta, GA, October 2006.

[125] A. Swaminathan, M. Wu, and K. J. R. Liu. Non-intrusive Forensic Analysis of Visual Sensors using Output Images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 401–404, Toulouse, France, May 2006.

[126] A. Swaminathan, M. Wu, and K. J. R. Liu. A Component Estimation Framework for Information Forensics. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 397–400, Crete, Greece, October 2007.

[127] A. Swaminathan, M. Wu, and K. J. R. Liu. Intrinsic Fingerprints for Image Authentication and Steganalysis. In *Proceedings of the SPIE Conference on*

Security, Stegonography, and Watermarking of Multimedia Contents, volume 6505, page 65051J, San Jose, CA, February 2007.

[128] A. Swaminathan, M. Wu, and K. J. R. Liu. Non-Intrusive Component Forensics of Visual Sensors using Output Images. *IEEE Trans. on Information Forensics and Security*, 2(1):91–106, March 2007.

[129] A. Swaminathan, M. Wu, and K. J. R. Liu. Optimization of Input Pattern for Semi Non-Intrusive Component Forensics of Digital Cameras. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 225–228, Honolulu, HI, April 2007.

[130] A. Swaminathan, M. Wu, and K. J. R. Liu. A Pattern Classification Framework for Theoretical Analysis of Component Forensics. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1665–1668, Las Vegas, NV, March 2008.

[131] A. Swaminathan, M. Wu, and K. J. R. Liu. Component Forensics: Theory, Methodologies, and Applications. *IEEE Signal Processing Magazine*, under review, July 2008.

[132] A. Swaminathan, M. Wu, and K. J. R. Liu. Digital Image Forensics via Intrinsic Fingerprints. *IEEE Trans. on Information Forensics and Security*, 3(1):101–117, March 2008.

[133] A. Swaminathan, M. Wu, and K. J. R. Liu. Semi Non-Intrusive Forensics. to be submitted, August 2008.

[134] A. Swaminathan, M. Wu, and K. J. R. Liu. Theoretical Analysis of Component Forensics. to be submitted, August 2008.

[135] M-J. Tsai, C-L. Lai, and J. Liu. Camera/ Mobile Phone Source Identification for Digital Forensics. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 221–224, Honolulu, HI, April 2007.

[136] M-J. Tsai and G-H. Wu. Using Image Features to Identify Camera Sources. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 297–300, Toulouse, France, May 2006.

[137] C. F. van Loan. *Introduction to Scientific Computing*. Prentice Hall, 2005.

[138] R. Venkatesan, S-M. Koon, M. H. Jakubowski, and P. Moulin. Robust Image Hashing. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 664–666, Vancouver, Canada, September 2000.

[139] E. A. Wan. Neural Network Classification: A Bayesian Interpretation. *IEEE Trans. on Neural Networks*, 1(4):303–305, December 1990.

[140] C-C. Weng, H. Chen, and C-S. Fuh. A Novel Automatic White Balance Method for Digital Still Cameras. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 4, pages 3801–3804, Kobe, Japan, May 2005.

[141] A. Westfeld. F5, Software available at `wwwwrn.inf.tu-dresden.de/westfeld/f5`.

[142] A. Westfeld. F5–A Steganographic Algorithm: High Capacity Despite Better Steganalysis. In *Proceedings of the Information Hiding Workshop and Lecture Notes in Computer Science*, volume 2137, pages 289–302, Pittsburgh, PA, Apr 2001.

[143] A. Westfeld and A. Pfitzmann. Attacks on Steganographic Systems. In *Proceedings of the Information Hiding Workshop and Lecture Notes in Computer Science*, volume 1768, pages 61–76, Dresden, Germany, September 1999.

[144] J. L. Wong, D. Kirovski, and M. Potkonjak. Computational Forensic Techniques for Intellectual Property Protection. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(6):987–994, June 2004.

[145] C. W. Wu. On the Design of Content-Based Multimedia Authentication Systems. *IEEE Trans. on Multimedia*, 4(3):385–393, September 2002.

[146] M. Wu and B. Liu. *Multimedia Data Hiding*. Springer Verlag, 2003.

[147] M. Wu, Y. Mao, and A. Swaminathan. A Signal Processing and Randomization Perspective of Robust and Secure Image Hashing. In *IEEE workshop on Statistical Signal Processing (SSP)*, pages 166–170, Madison, WI, August 2007.

[148] T-F. Wu, C-J. Lin, and R. C. Weng. Probability Estimates for Multi-class Classification by Pairwise Coupling. *The Journal of Machine Learning Research*, 5:975–1005, August 2004.

[149] A. Wyner and J. Ziv. The Rate-Distortion Function for Source Coding with Side Information at the Decoder. *IEEE Trans. on Information Theory*, 22(1):1–10, January 1976.

[150] F. Xiao, J. E. Farell, J. M. DiCarlo, and B. A. Wandell. Preferred Color Spaces for White Balancing. In *Proceedings of the SPIE Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications IV*, volume 5017, pages 342–350, Santa Clara, CA, January 2003.

[151] L. Xie, G. R. Arce, and R. F. Graveman. Approximate Image Message Authentication Codes. *IEEE Trans. on Multimedia*, 3(2):242–252, June 2001.