Statistical Image Source Model Identification and Forgery Detection

Cao Hong

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2010

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Date

Cao Hong

This thesis is dedicated to my wife, Anna Ong Xiao Fen,

for her love and support

and to my parents, Cao Ruigang and An Junfeng,

for their love and sacrifices.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my thesis supervisor, Prof. Alex Kot Chichung for his enormous patience, invaluable insight, his tough yet constructive criticisms, and for his continuous support and guidance on my research work over many years. This thesis can hardly be a success without him and his influence can be found everywhere. While working with him, I have gained tremendously both the technical knowledge and the way that original research work shall be done. I have been particularly impressed by his sharp eyes in finding the key underlying problems, his excellent guiding skills to let me see many issues with my own eyes, his critical out-of-box thinking skills, his thoroughness and diligence in revising every word and sentence in my research write-ups, his long-lasting passion and strict attitude toward scientific research work. I am thankful that he has set up a good example for me and the legacy he passed on me will surely benefit the rest of my life.

I would like to thank my thesis panel members, Assoc. Prof. Tan Yap Peng, Assist. Prof. Pina Marziliano and Assoc. Prof. Annamalai B. Premkumar for their valuable comments and support during my PhD qualifying examination. I would also like to thank Assoc. Prof. Xue Ping and Dr Chang Lanlan for kindly sharing with me their MATLAB codes of many demosaicing algorithms used in this study. I wish to thank all the professors in school of EEE who have taught me in various graduate courses or have shared with me their own research experiences and life stories. Particularly I want to thank Assoc. Prof. Bi Guoan, from whom I gained substantially in advanced signal processing course. I am also thankful to visiting Prof. Alfred M. Bruckstein, whose in-

depth image analysis course opens up my eyes to see the beauty of science. Besides professors, I also wish to extend my special appreciation to a number of my colleagues including Dr Yang Huijuan, who provided valuable comments on my write-ups during my early PhD stage and Dr Yap Pew Thian, from whom I was inspired to learn a lot of knowledge and skills related with cameras and photos. I also enjoyed the time that I spent with my other colleagues in Center for Information Security including Ren Jianfen, Dr Wee Chong Yaw and with my fellow graduate students including Li Sheng, Fan Jiayuan, Chen Mo, Geng Cong, Tan Guoxian, Chen Changsheng and Liu Siyuan. I am indebted to several undergraduate students, who have worked closely with me in exploring various image forensics problems in their final year projects, namely Lee Li Wah, Wai Chin Seng, Sai Choong Han, Monica Enescu, Elena Ovreiu and Chua Li Fu. Last but not the least, I am grateful to Stewart Chu, who has patiently provided me with administrative support and various resources on my many tedious requests. I also wish to thank Shavonne Tee, Agnes Tan, Phebe Tjahjono and Jean Choo Soo Ling for their consistent administrative support.

This thesis is dedicated to my wife, Anna Ong Xiao Fen, for her encouragement, understanding and continuous support. This thesis is also dedicated to my parents, Cai Ruigang and An Junfeng, for their love and sacrifices in bringing me up healthy and well-educated especially in a financially tight environment. I also appreciate the emotional support from my brother, Cao Lu, and his wife, Cao Xianzi. I want to thank my cousin, Assoc. Prof. Gao Hongwei, for helping me better understand the world research system and the world economy.

Finally, to everyone who has contributed to this thesis in one way or another, I express my sincere gratitude.

Abstract

Advances of digital technology have given birth to numerous unprecedented tools, which make image forgery easier than ever. To restore the traditional trustworthiness on digital photos, image forensics analyses that can reliably tell the origin, integrity and authenticity of a given image are urgently needed. In this thesis, we propose several new image forensics tools for: 1) Accurate detection of image demosaicing regularity as a general type of image forensics features; 2) Identification of various common image sources including digital still camera models, RAW conversion tools and the low-end mobile camera models; 3) Universal detection of a wide range of common image tampering and 4) Prevention of the image recapturing threat. These forensics tools help expose common image forgeries, especially those easy-to-make forgeries, which can hardly be seen directly by human eyes. The common theme behind our proposed forensics tools is through statistical detection of some intrinsic image regularity or tampering anomalies. Our tools are not constrained by the strict end-to-end protocol requirement such as prior image hash computation or prior information hiding; hence have bright application prospect. Advanced pattern classification techniques including feature reduction techniques and nonlinear classification methods are employed to achieve extremely good and better forensics performances than state-of-the-arts forensics methods based on large-scale experimental tests. In the universal image tampering detection framework, we have also proposed a novel FusionBoost learning to combine a set of lightweight probabilistic tampering detectors into a strong ensemble

tampering detector. Experimental results demonstrate its competency over the conventional boosting algorithms or fusion methods.

Contents

List of Figures

List of Tables

List of Abbreviations

1NNK	First Nearest Neighbor Classifier
AdaBoost	Adaptive Boosting Algorithm
AWGN	Additive White Gaussian Noise
BA	Brightness Adjustment
BS	Binary Similarity Features
CCD	Charge-Coupled Device Sensor
CF	Color Features
CFA	Color Filter Array
CG	Computer Graphics
CIC	Color Interpolation Coefficients Features
CMOS	Complementary Metal-Oxide Semiconductor
CMY	Cyan, Magenta and Yellow
CMYG	Cyan, Magenta, Yellow and Green
CST	Color Space Transformation
CV	Cross Validation
DC	Direct Current
DSC	Digital Still Camera
DSLR	Digital Single Lens Reflex
EC	Error Cumulants Features
EE	Edge Enhancement
EER	Equal Error Rate

EM	Expectation Maximization
EMRC	Expectation Maximization Reverse Classification
ERE	Eigenfeature Regularization and Extraction
EXIF	Exchangeable Image File Format
FAR	False Acceptance Rate
FRR	False Rejection Rate
FBI	Federal Bureau of Investigation
GC	Gamma Correction
GF	Gaussian Filtering
GHz	Giga Hertz
HE	Histogram Equalization
IQM	Image Quality Metrics Features
IS	Internet Source
JPEG	Joint Photographic Experts Group
LBP	Local Binary Pattern
LCD	Liquid Crystal Display
LDA	Linear Discriminant Analysis
MF	Median Filtering
MSWS	Multi-Scale Wavelet Statistics
NGS	Normalized Group Sizes features
NPPA	National Press Photographer Association
NStats	Noise Statistics features
OPS	One-Pixel Shift
P4	Pentium 4
PC	Personal Computer
PCA	Principal Component Analysis
PRCG	Photorealistic Computer Graphics
PRNU	Photo-Response Non-Uniformity
PSNR	Peak Signal to Noise Ratio
PSVM	Probabilistic Support Vector Machine
QT	Quantization
RBF	Radial Basis Function
ROC	Receiver Operating Characteristics

RS	Reliable Source
RT	Rotation
SA	Spatial Averaging
SCA	Scaling
SFFS	Sequential Forward Floating Search
SVM	Support Vector Machine
TIFF	Tagged Image File Format
UM	Unsharp Mask
US	United States
USA	United States of America
WB	White Balancing
WF	Wiener Filtering
WS	Wavelet statistics features
WT	Weights Features

Chapter 1 Introduction

1.1 Background

Cameras are traditionally regarded as trustworthy devices to capture the real-world events with no bias and the acquired photos often represent the truth [1]. The traditional trustworthiness on a photo largely relies on the notable difficulties to modify its content. For instance, in the past, altering a film-based photo had to be made through the so-called darkroom tricks by photo specialists, which is costly, time-consuming and moreover the extent of forgery is often very limited. In contrast, the recent advances of digital technology have given birth to many powerful devices and digital tools such as high-performance computers, high-resolution digital cameras, high-definition LCD displays, high-speed Internet connections and highly advanced image editing tools. These commercial tools are made available to a large number of ordinary people. While people enjoy the great conveniences of using these tools, a serious question would be asked like "Can we still trust what we see in this digital age?" With the popular Photoshop software, almost everything on a photo can be deliberately modified to deceive others the truth. Most of the time, human eyes can hardly differentiate whether a photo is genuine or it has been maliciously tampered. Consequently, increasing fraudulent cases involving photo forgeries have appeared in recent years and we highlight two recent impactful cases in Fig. 1-1. In the first case, Spanish politician Gasper Llamazares was shocked to find out that the FBI of USA had used one of his photos for creating a digitally altered image to show publicly how the most wanted terrorist, Osama Bin Laden, might look in FBI's Rewards of Justice Website. FBI

Fig. 1-1 Two Recent Impactful Photo Forgery Cases; Case 1: The Photo Published by USA Government in a Poster of the Most Wanted Terrorist, Osama Bin Laden, is Found to Be a PhotoShopped Composite of the Internet Image of a Spanish Politician, Gasper Llamazares, in (b) and A Osama Bin Laden's Photo (1998) in (c) [2]; Case 2: Comparison of the Original Photo of Israel Cabinet in (d) and the Forgery Photo in (e) Appearing in a Newspaper [3], Where the Tampered Areas are Highlighted

admitted that they had used the facial features of an image they found on Internet and applied some "cutting edge" technology to show how Bin Laden might look now. Such an incident has well embarrassed the US government and the Spaniard condemned the "low level" of US intelligent services indicating that the digitally altered photo had seriously threatened his own safety as he could be wrongly identified as the most wanted terrorist. In the second case, the inaugural photo of the Israel cabinet were digitally tampered, where two female cabinet members were replaced by two male ministers, in the Daily Yated Neeman newspaper to cater for the taste of ultra-orthodox Jewish readers. This sparked controversy of gender discrimination in Israel society. More famous photo forgery examples can be found in [4-7] and these photos were mostly published in the world-renowned newspapers and magazines.

As journalists we believe the guiding principle of our profession is accuracy; therefore, we believe it is wrong to alter the content of a photograph in any way that it deceives the public.

As photojournalists, we have the responsibility to document society and to preserve its image as a matter of historical record. It is clear that the emerging electronic technologies provide new challenges to the integrity of photographic images. The technology enables the manipulation of the content of an image in such a way that the change is virtually undetectable. In light of this, we, the National Press Photographers Association, reaffirm the basis of our ethics: Accurate representation is the benchmark of our profession.

We believe photojournalistic guidelines for fair and accurate reporting should be the criteria for judging what may be done electronically to a photograph. Altering the editorial content of photograph, in any degree is a breach of the ethical standards recognized by the NPPA.

Fig. 1-2 Guidelines for News Photograph from NPPA [1]

The ability to alter photos through electronic manipulation has raised a host of legal, moral, and ethical issues. The United States (US) defense officials have often been accused of exaggeration and manipulation to show the military in the best light [1]. Concerned that the possible photo manipulations are detrimental to the image of the Department of Defense, J.M. Deutch, as Deputy Secretary of Defense, issued strong directive on using the imagery for publication. The directive states that imagery must be complete, timely, and above all, highly accurate. “Anything that weakens or casts doubt on the credibility of this imagery within or outside of the Department of Defense will not be tolerated.” Similar strict standards have also been set by leading newspapers and magazines. However, there are still concerns about possible abuses since it is generally known that news photos have been manipulated by the editors. Some news agencies feel that it is permissible to “clean up” a photo as long as that doesn’t change the integrity of the image. Other organizations maintain the right to “clean up” a photography that appears cluttered or messy by removing small elements in the photograph that are “journalistically irrelevant”. Though some image altering is allowed, it is the editors’ responsibility to ensure the manipulations made do not undermine the accuracy of the photos and frequently the editor would need to keep a copy of the original unaltered photo. The US National Press Photographer Association (NPPA) has also issued a call of guidelines for news photos, which states the principle in Fig. 1-2. This principle clearly

states the importance of maintaining the accuracy of the photos and electronic manipulation of the image content is an unethical behavior for photojournalists.

While photos are still popularly used as supporting evidences and historical records in growing number and wide range of applications from journalist reporting, police investigation, law enforcement, insurance claims, medical and dental examinations, military, museum and consumer photography, scientific means that can tell their origin and verify their authenticity and integrity are in urgent needs. This urgency is stimulated by several new trends: 1) Proliferation of digital cameras and other multimedia acquisition devices is fast due to the improving photographic technology and the reducing cost. Digital cameras equipped on mobile handsets now have become a necessity feature [8] for mobile phone users; 2) Photo sharing on internet has become increasingly popular. Media sharing and social network websites such as *Flickr* and *Facebook* promote users to upload and share their own photos. These provide common platforms for the forgery photos to spread quickly to make great social and political impacts; 3) A significant portion of photos from the public has good news value and is sourced by various news agencies for making journalist reports [9, 10]. However, one big challenge [9] is how to authenticate the photo contents from the public, which is generally considered an unreliable source; 4) Following the human stem-cell fake [5], which involves obvious image forgeries, scientific journal editors [11] have been actively seeking efficient forensics tools, which detect the deliberate image tweaks.

1.2 Image Forgery Categories

We use the term “image forgery” to broadly refer to all the fraudulent images which are deliberately created for the purposes of misguidance or deceiving others. In Fig. 1-3, we broadly classify image forgery into three distinct categories, altering an existing image, creation from scratch and scenery forgery.

1. **Altering an existing image**: Supported with today’s high-performance computers and state-of-the-art image editing tools, this forgery category has become the easiest and the most convenient means to fake images. Depending on the way forgery is made, we further divide this category into six types, including object based, splicing

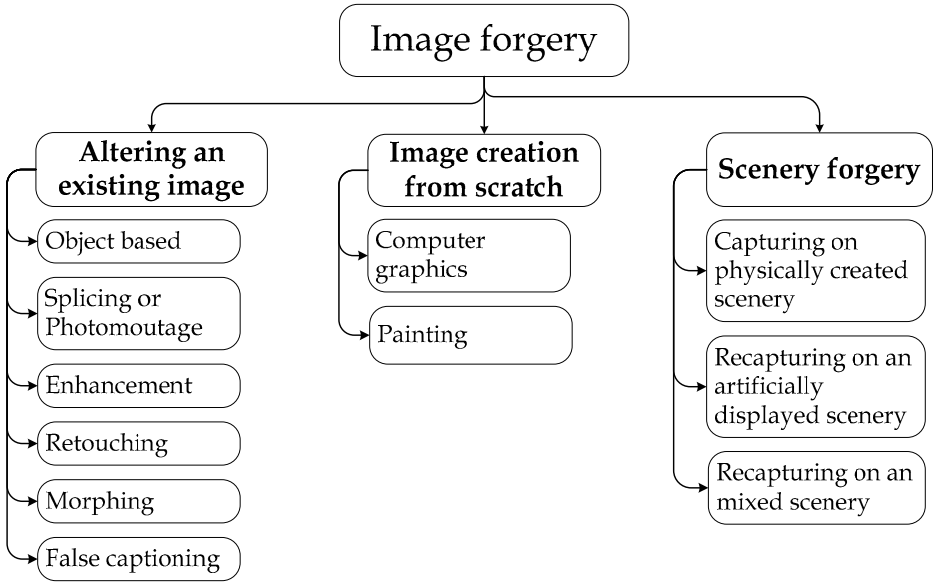
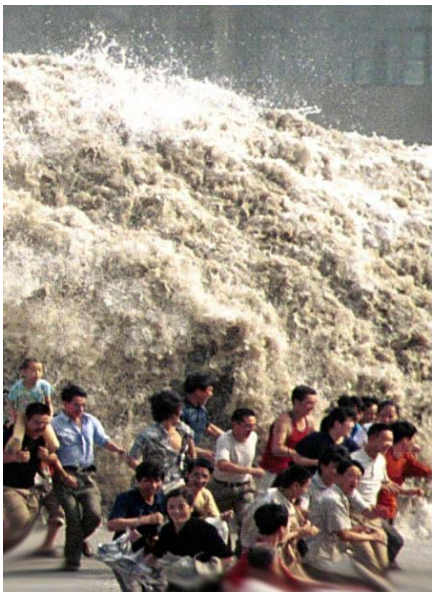


Fig. 1-3 Image Forgery Categories

or photomontage, enhancement, retouching, morphing and false captioning as shown in Fig. 1-3. The object-based type refers to the object manipulation related operations such as object insertion, deletion, duplication and changing an object's attributes including size, shape, color, orientation and so on. The splicing or photomontage forgery refers to the forgery of compositing several images into one. Enhancement here refers to the broad class of global image manipulation operations such as denoising, sharpening or blurring, adjustment of hue, contrast and brightness and histogram modification. Retouching, on another hand, refers to some local small-scale image forgery operations, such as correction of red-eyes, slight modification on an edge curvature, covering up moles and removing hairs on the skin. Morphing is a common technique to smoothly transform a source entity into a target entity. Image morphing typically involves detection and matching of the salient feature points from a source image to the target image. To create image forgery using morphing, for example, one can modify the source face image of a normal human being towards a target image of an alien's face by using the morphing technique in [12]. False captioning is a special type of forgery, in which, the description of an image or the metadata tags associated with image are modified instead the image itself. Such kind of image forgery is frequently seen on newspapers and magazines and shown in Fig. 1-4 is a well-known example.



(a)

(b)

Fig. 1-4 A Picture Taken in Qian Tang Jiang River, Hanzhou, China in (a) is Cropped and Falsely Captioned in a Newspaper as an Indian Ocean Tsunami Picture in (b) Taken in Thailand in December 2004 [6]

2. **Image forgery creation from scratch**: Artificially creating an image from scratch is generally very difficult but is still made possible with support of start-of-the-art computer graphic and electronic painting tools. With computer graphic tools like *Maya* and *3ds Max*, an image forger can first construct some three-dimensional (3D) polygonal shape models and then augment them with details including color, texture and illumination. The augmented 3D model is further photorealistically rendered by a virtual camera to create the final syntactic image. Painting is a traditional way to create an image from scratch. Supported with high-resolution and high-data rate Tablet LCDs or Tablet PCs, one can conveniently paint an object electronically, e.g. a car plate, onto the LCD screen using tools like PaintShop Pro and CorelDraw. The electronic painting enables layered processing and easy correction and this helps skilled people to accomplish a painting task with better quality and less time. However, to paint an image with a feeling of photorealism in general is still a very difficult task, which requires tremendous skills and photographic knowledge.
3. **Scenery forgery**: Contrary to the direct alteration on an existing image, one can make indirect forgery by capturing on artificially created scenery. The types of artificial scenes can be classified into the physical scenes, artificially displayed scenes and the mixture of both types of scenes. Similar to what is done in the movie

production, one can physically set up a scene even with professional human actors/actresses involved. However, since creating such a physical scene by itself can be a very expensive and even impossible task in many occasions, forgery with a physical scene only is generally very difficult for the ordinary people. On another hand, the artificially displayed scenes are relatively easy to create. The advances in display technology have brought us the ubiquitous high-resolution display devices, high-quality color printers and projections. These devices can be easily used to display fake scenes, e.g. a manipulated photo, so that the forged scene can be recaptured. Note that the recaptured photos are still the direct output from a camera where no obvious tampering traces shall exist. Alternatively, one can fake a scene by mixing artificially displayed objects into the physical environment in order to deceive others.

Very often in a practical scenario, an image forger simultaneously employs a mixture of the above forgery techniques. For example, when a montage image is made, the manipulator needs to delicately cut a selected object from one existing image, rotate, resize the object and adjust its color hue and illumination property before the object is transplanted onto another photo. To make it look better, the relatively distinctive splicing boundary is often softened through blurring, the object edges are smoothed and the tiny image defects are removed through using retouching tools when the picture is zoomed into a large scale. At last, enhancement techniques such as sharpening and histogram modifications are popularly used to improve the overall visual perspectives.

1.3 Related Prior Works

In this section, we review some existing techniques related with image forensics based on the following organization: Section 1.3.1 describes two categories of image forensics, namely the active and the passive forensics with a highlight of the typical camera processing pipeline. Section 1.3.2 enumerates the current passive forensics issues and discusses their general solutions and the features used. As our thesis focuses on source model identification and forgery detection, Section 1.3.3 and Section 1.3.4 review the existing techniques in image source identification and forgery detection,

respectively. We also note that the forensics challenges are often formulated and solved as pattern classification tasks, where pattern classification techniques play important roles for achieving good performances. Section 1.3.5 reviews the commonly used pattern classification techniques for image forensics. In Section 1.3.6, we discuss some emerging attacks on some tampering forensics techniques, namely tamper hiding techniques.

1.3.1 Active and Passive Forensics

To restore the public "trust" towards digital photos and to alert deliberate tampering, image forensics has attracted considerable attention in the recent decade. The proposed approaches generally fall into two broad categories, the *active* and the *passive* categories. The *active* approaches commonly require pre-computation of the some fragile image property, e.g. a cryptographic hash, or prior insertion of some protection data through information hiding techniques before the images are sent through an unreliable public channel. Upon receiving these images, forensics conclusions can be made by comparing the recomputed image property or the extracted data with their original versions. For instance, two early works actively modify the camera structure to build the "trustworthy" or the "secure" digital camera to protect integrity of the photos. In [15], an encrypted digital signature file is separately generated for each photo in a "trustworthy camera" for authentication purposes. In [16], a watermark with iris biometric data inclusive is embedded in a "secure camera" and the extracted watermark can be used for both photo integrity verification and camera taker identification. More *active* approaches can be found in [17] for forensics applications such as content authentication and digital rights management for various multimedia types. A recent approach in [18] also proposes to protect integrity of the emerging electronic ink data through a lossless data embedding technique.

Though the *active* approaches can be highly effective to secure photos from malicious tampering, the strict requirement on the cooperative end-to-end protocols is hardly met in most of today's digital cameras. On the other hand, the *passive* approaches do not impose such constraints and they are readily applicable to a wide range of image authentication scenarios. The common philosophy behind passive image forensics is that

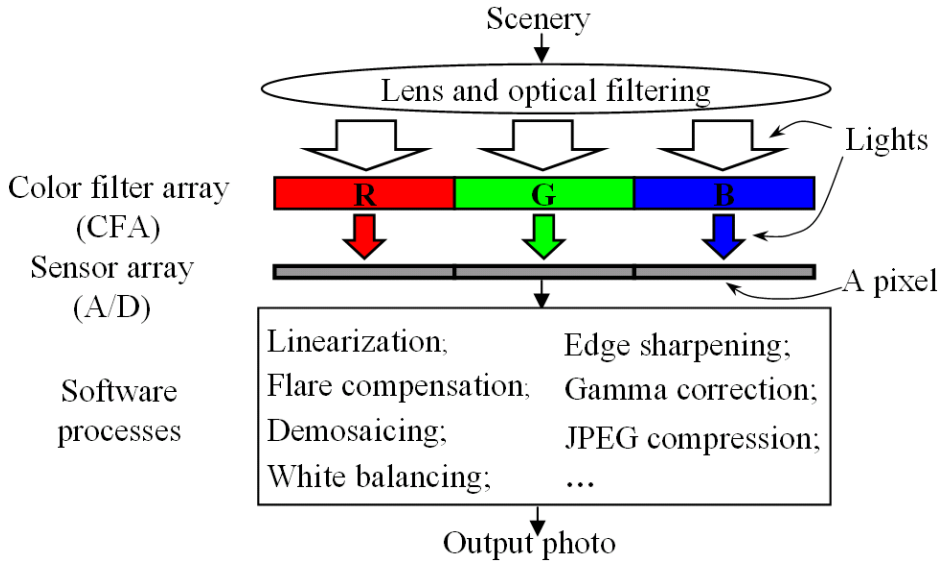


Fig. 1-5 A Typical Image Processing Pipeline [13, 14]

the photos acquired through image acquisition devices are mainly natural images, which occupy only a very small and highly regularized space in the entire high-dimensional image space. Tampering a photo highly likely disturbs some existing image regularity, introduces new artifacts and lead to many forms of inconsistencies. If detected, these inconsistencies can be used as tell-tale signs to expose the forgery.

The passive image forensics research often requires tremendous knowledge on the image creation processes inside the common acquisition devices. Though the different devices can differ significantly in their internal processing, a typical skeleton for a commercial single-sensor digital still camera (DSC) is shown in Fig. 1-5. When light photons enter into a camera, they firstly pass through a lens system together with several optical filters including anti-aliasing filters (blurring filter) and infrared rejection filters. Before the photons arrive at the sensor array, the light is further filtered by an array of tiny color filters, known as a color filter array (CFA), where one color filter is placed on top of each sensor. Since both the commonly-used CCD and CMOS sensors can only sense the luminance, color filtering allows each sensor to measure a specific color at each photo site (pixel). After the sensor data are captured, several compensation processes like color calibration, linearization, dark current compensation and flare compensation may take place as pre-demosaicing processes [14]. Since only one color is available at each pixel, demosaicing here is a reconstruction process that interpolates the

missing color samples in order to generate a three-color image. The reconstructed image after demosaicing further goes through possible processes like color transformation, white balancing [19], color-artifact removal, edge-enhancement, coring and lossy JPEG compression [20, 21] before the end photo is available.

1.3.2 Current Passive Forensics Issues and General Solutions

The good application prospects of passive image forensics have attracted considerable attention in recent research. Past researchers have mainly addressed the following photo-authentication related challenges:

1. **Image source related** [22-61]: Which individual device, what sensor type, model and brand is the device used in capturing a given image? Is an image acquired with a device as claimed? Given a set of images, can we cluster the images according to their sources?
2. **Tampering discovery** [50, 67-98]: Has a given image been tampered and where is the tampered region?
3. **Steganalysis** [99-101]: Has an image been concealed with a secret message?
4. **Recovery of processing history** [102-105]: What processes have an image gone through? What are the parameters used?
5. **Image recapturing identification** [63-64, 106]: Is a given image captured on real scenery or on artificially created scenery using the modern display technology?
6. **Computer graphic image identification** [62-66]: Is a given image acquired using a camera device or artificially created using photorealistic computer graphic (PRCG) software?
7. **Device temporal forensics** [107]: When is a given image captured? How can we order a sequence of images according to their capturing time?

Based on the features used, the possible solutions to the above image forensics topics generally fall into two broad categories: 1) through detection of tell-tale artifacts introduced by common tampering operations; 2) through detection of certain intrinsic image regularities and their inconsistencies. In the first category, previous works have detected the tampering artifacts due to copy-move or region duplication [67, 72, 81, 93,

98], resampling (e.g. rotation and scaling) [72, 76], splicing [69-70, 84, 97], double JPEG compression [72, 78, 83, 88, 91], etc. Since these artifacts rarely exist in an untampered image, their successful detection can be used as evidences of tampering. The second category is based on the facts that many types of image regularities exist in an image and these regularities can be caused by the imperfection of certain hardware or some software process of the camera system. Some regularities are fragile towards common image tampering operations hence can be detected as a device signature for a wide range of forensics applications. Previous works have attempted to detect numerous types of image regularities and we classify them into the four categories below according to the typical camera processing pipeline in Fig. 1-5.

- *Optical regularities* include illumination [74, 85-87], lens radial distortion [29-32], chromatic aberration [47, 79, 85] and blurring [73, 89], which are caused by the physical law on light transportation, optical filtering and imperfection of the lens system;
- *Sensor imperfections* including sensor noise pattern [22-25, 34, 37, 40, 42, 44, 50, 52, 55-56, 58, 60-61, 80, 107] and statistics [38, 41, 43, 53, 57], dust pattern [39, 51, 59] and camera response function [104], which are introduced when light signal is converted into digital signal;
- *Processing regularities* including demosaicing [27-28, 33, 45, 48, 54, 57, 75, 90, 94], white balancing, gamma correction [102], JPEG compression etc., which are introduced by the in-camera software processes;
- *Other statistical regularities* including natural image statistics [77, 101, 62-66], color features [26, 35, 65] and image quality metrics [26, 36, 68, 77, 99], which characterize the general statistics of natural images.

The different regularities are associated with different origins and their detections are useful in various forensics tasks. Detection of the optical regularity can be readily applied to expose image forgeries even without the knowledge of image sources. However, reliable and automated detection of the optical image regularities by itself is a challenging problem especially in complex lighting environment and in the presence of image noises and processing distortions. The sensor noise pattern or dust pattern are features associated with the sensor chip inside a camera. Detection of such patterns is

useful to identify individual cameras and to expose image integrity tampering. Detection of processing regularities helps to reverse engineer the digital technologies and discover the adopted processing parameters inside a camera. Very often, the detected parameters can also be used as forensics features to characterize a group of cameras sharing a similar image processing pipeline, e.g. cameras of the same model. The performances of the above methods generally depend on both the feature types used and appropriateness of the detection methods. For more about the current issues and the passive forensics solutions, one can also refer to several recent surveys in [108-113].

1.3.3 Image Source Identification

The research on image source identification investigates on the techniques to identify the image acquisition device or its certain attribute, e.g. the device model, for a given image. Depending on the research outcomes, two main branches of image source identification are image source model identification and individual source device identification. Below, we review the techniques for each category:

Image Source Model Identification

Since image acquisition devices of the same model often share identical software processing modules and very similar hardware components, the different source models can be identified by extracting features that are intrinsic and unique to each device model. The commonly used features are the processing regularities associated with some software modules, e.g. demosaicing regularity and color features, or the regularities, e.g. noise statistics, image quality metrics, lens distortions, which are associated with some attributes of the hardware, e.g. quality of the sensor and distortion characteristics of the lens system.

Demosaicing regularity is popularly used in source model identification as the choice of CFA and demosaicing algorithm is usually fixed for a given device model but likely differs for different models. In [75], Popescu *et al.* proposed an expectation maximization (EM) algorithm for estimating a set of 24 interpolation filter weights, which characterize the pixel correlations in the red, green and blue channels. By using these weights as features for a linear classifier, an average accuracy of 97% was

achieved in distinguishing 28 pairs of demosaicing algorithms, where the minimal pairwise accuracy is 88%. Bayram *et al.* [27, 28] extended this EM algorithm by assuming a larger 5×5 interpolation window, and analyzed patterns of periodicity in row-averaged second-order derivatives in the non-smooth and smooth image parts, respectively. The filter weights and the periodicity property are used as features in constructing support vector machine (SVM) classifier. An average accuracy of 95.9% was achieved for identifying three camera models. Based on a quadratic pixel correlation model, Long *et al.* [33] proposed to compute the normalized 13×13 pixel correlation coefficients for each color channel as features to train a neural network classifier for identifying different source camera models. With a majority-vote fusion, a close to 100% accuracy was reported for identifying four commercial cameras of different models in the presence of 0-5% rejected or undetermined cases. Swaminathan *et al.* [45] estimate the underlying demosaicing parameters for non-intrusive component forensics analysis on different camera models. Based on an intra-color channel correlation model, this work used a total least square solution to estimate the camera demosaicing parameters in three color channels and in three types of image regions including the horizontal-edge region, the vertical-edge region and the smooth region. The estimated demosaicing parameters are used as features in identifying 19 cameras of different models. With 200 images per camera model, these features achieved an average identification accuracy of 86%.

Other statistical regularities have also been used for camera model identification. Kharrazi *et al.* [26] proposed 34 features including color features, image quality metrics and wavelet coefficient statistics. With a SVM classifier, these features yield an accuracy of 88% in identifying five camera models, where three are from Canon brand. Based on a second-order lens radial distortion model, Choi *et al.* [29-32] proposed a method to estimate the lens radial distortion parameters through measuring the distortions on extracted straight-line edge segments. By using the estimated parameters as features together with the Kharrazi's features in [26], an accuracy of 89.1% is achieved in identifying five cameras models based on a fixed optical zooming. However, the accuracy of using the distortion parameters alone can be severely affected if the optical zooming of the lens is varied. Van *et al.* [47] proposed to estimate the distortion parameters associated with the lens's lateral chromatic aberration for mobile camera

identification. By using these parameters as features for SVM classification, an identification accuracy of 92.2% was achieved for three different mobile phone models. The work also provided some results showing that the features cannot distinguish two individual mobile cameras of the same model. Celiktutan *et al.* [49] combined three sets of features including binary similarity measures, image quality metrics and wavelet coefficients statistics for identification of cell-phone camera models. With feature-level fusion and sequential forward floating search feature selection, it reported an average identification accuracy of 95.1% for sixteen cell-phone models. This work has also tested score-level fusion strategies with different rules, where a maximal accuracy of 97.5% is reported based on the product rule. For scanner model identification, Gou *et al.* [41] proposed to compute a set of 60 sensor noise statistics features. With a SVM classifier, 90% accuracy was achieved in identifying seven scanner models. McKay *et al.* [57] combined the noise statistics features in [41] together with color interpolation coefficients [45] for identifying different source types including computer graphics, digital still cameras, phone cameras and scanners and their source models. An average identification accuracy of 97.7% was achieved for five cell-phone camera models, 96.2% for four scanner models and 94.3% for four camera models.

Individual Source Device Identification

Identification of individual image acquisition devices, especially different device units within a same model, often requires extraction of some defects patterns as features that are intrinsic and unique to each individual device.

Based on twelve cameras from a brand called Trust, Geradts *et al.* [22] discovered that the defective pixels from the CCD image sensors often reside at the same places on the blank images from the same camera but at different places for different cameras. The pattern of these defective spots was suggested as a fingerprint for identifying individual cameras. Several follow-up works in [23-25] also showed that similar patterns of the defective spots could be observed in the images from video cameras, other CCD-sensor cameras and some CMOS-sensor cameras. Though these works [22-25] reported a remarkable phenomenon, they did not develop a viable method to extract this fingerprint in the ordinary non-blank photos to identify the source camera devices. Lucas *et al.* [34]

proposed to extract photo-response non-uniformity (PRNU) sensor pattern noises for individual camera identification. With nine cameras and 300 training photos from each camera, it shows that the synchronized PRNU noise residue patterns can be used to identify the camera sources based on a correlation detector with a close-to-zero false rejection rate (FRR) when the false acceptance rate (FAR) is fixed at 0.1%. In the follow-up work, Chen *et al.* [50] have improved the noise model, the preprocessing techniques for finding the PRNU noise pattern and the correlation detector used. As a result, fewer training photos are needed to obtain the reference PRNU pattern and the performances for individual camera identification are improved. For scanners, Khanna *et al.* [42] extended this methodology to detect the one-dimensional PRNU sensor noise pattern from scanned images. An average accuracy of 96% was achieved in identification of four different scanners. More variations of the PRNU pattern based methods for source device identification, fingerprinting or device-based clustering can be found in [44, 52, 55, 56, 58, 60, 61].

Besides the sensor noise patterns, dust and scratch patterns have also been suggested for individual device identification. Dirik *et al.* [39, 51] proposed a method to detect the sensor dust specks on images from digital single-lens reflex (DSLR) cameras based on match filtering with an empirical dust model and contour analysis. The detected dusk specks at different locations form the dust reference template for individual DSLR camera device identification. With 100 images from a testing source camera and 1000 images from eight other cameras, above 95% identification accuracies are achieved for three different test cameras by separating the two scenarios subject to the availability of the source DSLR camera. Dirik *et al.* [59] further extended this dust-pattern based method to scanner identification. Based on two test scanners, the scatter plots show that the correlation scores for a matched case are generally significantly larger than those of the unmatched cases.

Though good performances are reported for these defects-pattern based approaches for source identification at the precision of individual devices, these techniques require good pattern synchronization for computing the correlation scores. Moreover, a good number of training photos for a given test device, especially those with uniform and non-saturated scenery, are often needed for reliably generating the reference template. Also

due to the stochastic nature of the defect locations, the reference defect pattern learned for one device cannot be extended to the perform source identification for images acquired from another device, even if the two devices are close devices of the same model. On the other hand, the statistical regularities for source model identification are expected to have good generalization capability for a group of devices of the same model, but they are not expected to distinguish well the individual devices of the same model.

1.3.4 Forgery Detection

We refer forgery detection as a collection of techniques to expose the various types of image forgeries that we summarized in Fig. 1-3. As the primary goal of passive image forensics is to restore the trustworthiness on photos, reliable forgery detection is crucial for achieving this goal.

As discussed early, the most common forgeries are through altering or tampering an existing image. Therefore, tampering discovery or tampering forensics is the main category of techniques in forgery detection. Though tampering forensics and source identification address two separate forensics issues, we find that many tampering forensics solutions are closely linked with those for source identification. Here, we summarize these links: 1) Source identification can be used directly to address one type of tampering, where a forger modifies an image's metadata tags and falsely claim an image from one device as from another device; 2) Image tampering often involves mixing signals from multiple sources, which lead to source-dependant inconsistencies. These inconsistencies can be characterized either through using the source-identification related features or based on the outcomes of the source identification techniques on different image parts; 3) Many features used for source identification are found to be good to address the tampering forensics issue due to their fragile nature towards image tampering. 4) Source identification can serve as an initial step for tampering discovery. A forensics analyst can identify the image source first and then apply the corresponding source-dependent template to discover the local tampering.

Though source identification techniques can be extended to address the issues in tampering forensics, the tampering forensics approaches do not necessarily need to be related with source identification. Also for detecting other forms of image forgeries, e.g. scenery forgery and falsely claiming a PRCG image as a photo or vice versa, we consider the techniques for detecting the recaptured scenery and for detecting PRCG images as another category of forgery detection techniques. Below, we briefly review the forgery detection techniques in three broad categories, the source-related tampering forensics, the non-source-related tampering forensics and detection of unconventional images, such as recaptured scenery and PRCGs.

Source-Related Tampering Forensics

After achieving good source device identification results in [34], Lucas *et al.* [80] extended the PRNU pattern approach to discover local tampered region of interest (ROI). Through correlating the local PRNU patterns extracted from a test image with the synchronized reference PRNU patterns based on different sliding block shapes and sizes, the forged ROI is automatically determined and it shows relative reliable identification at a JPEG quality factor of 70. By further improving the PRNU model, Chen *et al.* [50] investigated on both identification of the source camera devices and verification of image integrity. Based on a sliding block size of 128×128, the correlation statistics for each pixel is measured and converted into a probabilistic score of tampering, which is subsequently used to determine whether a pixel is tampered. For JPEG quality factor 75, the integrity verification results based on 345 cut-and-paste forged Canon G2 images, where the forgery area limited to 228-512 pixels, shows that in 73% of the forgeries, at least 2/3 of the forged area is correctly identified, while in 21% of all the cases, more than 20% of the falsely detected pixels are available.

As discussed earlier, Popescu *et al.* [75] proposed to use an EM algorithm to estimate the interpolation filter weights, which can distinguish different pairs of demosaicing algorithms. As another outcome this approach, a probability map is generated, which exhibit periodical attributes and show unique localized peaks in Fourier transformed domain for different demosaicing algorithms. In a block-based manner, these frequency peaks' features show good results in determining whether CFA

interpolation traces exist in a local image patch. Swaminathan *et al.* [90] extended their non-intrusive component forensics work in [45] to tampering forensics based on the facts that image manipulation would alter the underlying demosaicing correlation. By formulating the extrinsic image manipulation as a linear time invariant filtering process, this work estimates the filter parameters through a recursive deconvolution technique. A similarity score is then used for detecting different forms of image manipulations. It reported 80-95% accuracy for detecting image manipulations such as linear average filtering, median filtering, additive noise, rotation, histogram equalization and resampling on images from a Canon Powershot A75 camera. Reasonably good results in terms of receiver operating characteristic (ROC) curves are also demonstrated in steganalysis and in detection of PRCGs, scanned images and cut-and-paste forgery. In another work, Dirik *et al.* [94] proposed to compute two demosaicing features, which are related with Bayer CFA pattern number and analysis of the interpolation noise, respectively, to detect image tampering. Close to 100% detection accuracies are reported for detecting five manipulations including blurring, downsizing, upsizing, rotation and JPEG recompression.

Bayram *et al.* [77] combine three sets of statistical forensics features including image quality metrics, binary similarity and wavelet statistics for image manipulation detection. Note that these three types of features have also been proposed for camera model identification [36, 49], steganalysis [99-101] and detection of PRCGs [62, 64]. Through sequential forward floating search feature selection and based on SVM classification, this work detects manipulations including scaling, rotation, contrast enhancement, brightness adjustment, blurring/sharpening and the combinations of these tampering types with different parameters. Based on 200 authentic images from a Canon Powershot S200 camera and 6000 manipulated images, the joint feature set reports the maximum of about 82% detection accuracy in a blind detection mode, where the tampering type and tampering parameters are unknown.

Johnson *et al.* [79] proposed to detect the inconsistent lateral chromatic aberration (LCA) as a tell-tale sign of image tampering. LCA is formed due to the varying refractive indexes of lens materials for different light wavelengths so that the light of different colors of cannot be perfectly focused. This causes misalignment in different

color channels, which are small near the optical center but become larger at the locations that are farther away from the optical center. Based on a linear LCA model, the method estimates the global model parameters, e.g. coordinates of the optical center and a scalar value, using an image registration technique that maximizes the mutual entropy between different color channels. By also estimating the LCA in localized image patches and comparing the local estimation with the global estimation, the blocks with large deviations in the LCA distortion orientations are deemed as the tampered blocks. As LCA can be corrected by the in-camera's software processing unit, the efficacy of using LCA still need to be evaluated in large-scale test in practical scenarios. Note that these LCA distortion parameters are also used as features to identify mobile camera models in [47] based on the same LCA model.

Non-Source-Related Tampering Forensics

The features used in these techniques include the optical regularity in the scenery or some anomalies associated with some specific tampering operations.

Johnson *et al.* [75] developed a tool to estimate the illuminating direction from a point light source on a single image. Based on the occluding boundaries, whose surface normal component in the *z*-axis is zero, this work employs a reduced Lambertian reflectance model to estimate the lighting direction in the *XY* image plane. With a small set of images including both syntactic and real images, it shows that the inconsistent lighting directions estimated from the different occluding boundaries can be a good tampering indicator in three different lighting scenarios, infinite light source, local light source and multiple light sources. Along this line, Johnson *et al.* have also extended this idea to other forensics scenarios, e.g. based on the specular highlights from human eyes [86] and in a complex lighting environment [87].

For methods that are associated with specific tampering types, Popescu *et al.* [76] proposed an EM algorithm to detect the presence of periodical pixel correlation and detect it as an evidence of image resampling. Ng *et al.* [69, 70] proposed a set of high-order statistical features to detect the presence of the sharp image discontinuities caused by image splicing. As copy-and-move forgery would introduce highly-correlated image

regions, Fridrich *et al.* [67] proposed an efficient searching algorithm using quantized DCT coefficients as features to search for the repeated image regions.

Since photos are popularly JPEG compressed by default, double JPEG compression can also be detected as a sign that an image has been potentially altered and resaved in JPEG format. Popescu *et al.* [71] observed that the second compression with a different quantization step would lead to the periodical artifacts in histogram of the DCT coefficients, which can be used as a tell-tale sign of double JPEG quantization. Fu *et al*. [83] further discovered that the first digits of JPEG DCT coefficients for a single JPEG compressed image closely follow a generalized Benford's law but not for the double JPEG compressed images. The violation of the generalized Benford's law is detected as an evidence of JPEG compression for more than one time. He *et al.* [78] proposed to recompress a JPEG photo with a high quality factor and identify the blocks that do not exhibit double-quantization effect as doctored blocks. Through analyzing the discrete cosine transform (DCT) coefficients, the probability for each block being doctored is estimated and this probability map helps a forensics analyst to visually identify the tampered image region. Luo *et al.* [81] proposed to measure the JPEG blocking artifacts for differentiating single-compressed and double-compressed image blocks.

Detection of Unconventional Images

For detection of PRCGs, Lyu *et al.* [62] proposed to use a set of 216 wavelet statistics features to characterize the statistics of natural images. With a SVM classifier and based on 40,000 photos and 6,000 PRCGs, it reported an accuracy of 66.8% to classify photos with a false-negative rate of 1.2%. Motivated by the different physical image generation processes for photos and PRCGs, Ng *et al.* [63] proposed a set of 192 geometry features and reported an overall classification accuracy of 83.5% based on 800 PRCGs and 2400 photos using a SVM classifier. To characterize several types of perceptional differences between photos and PRCGs, Chen *et al.* [65] proposed a set of low-level features related with color, ranked histogram, ranked region size, correlogram etc. Using an AdaBoost classifier and based on 36,000 photos and 35,000 PRCGs, it reported an average detection error rate of 5.5%.

For detecting recaptured images, Lyu [64] suggested computing 72 wavelet statistics features from the gray image plane to distinguish color photos and the printed-and-scanned photos. With 1000 photos and 200 such recaptured photos, it reported 99.5% test classification rate with a 0.2% false negative rate. Since image recapturing is often performed on a planar surface, Yu *et al.* [106] proposed to study the specularity distribution for detecting recaptured photos as the specularity is known to be uniform on a planar surface. Based on dichromatic reflectance model, several analyses show that the specularity distribution tends to be Laplacian-alike for natural photos while those for recaptured photos tend to be Rayleigh-alike. Though the efficacy of this method is demonstrated using the fake recaptured tiger photo in [149, 150], its further evaluation based on large-scale tests is still needed. Ng *et al.* [63] also reported 96.6% and 97.2% accuracies to classify recaptured PRCGs on LCD screens from the real PRCGs based on geometry features in [63] and wavelet statistics features in [62], respectively.

1.3.5 Forensics Pattern Classification Techniques

Since most image forensics problems can be readily formulated as pattern classification tasks, employment of suitable pattern classification and related techniques plays an important role for achieving good forensics performances. Below we discuss the feature reduction, pattern classification and classifier fusion techniques used in prior forensics works.

Feature Reduction Techniques

High feature dimensionality can easily cause increased computational complexity and incurs long classifier training time and slow forensics responses. The relatively small size of training data as compared with feature dimension could also lead to insufficient classifier training with degraded test performance. It is often desirable to substantially reduce the feature dimensionality before the reduced features are used in forensics analysis. The feature dimension reduction can be achieved using either subspace transformation or feature selection techniques.

The subspace approaches typically search for a compact set of projection axes or a subspace from the entire high-dimensional image or feature space. Through projecting

the high-dimensional feature data onto each axis, a new feature is generated as a linear combination of the old features. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two commonly used subspace methods in many pattern classification tasks. With an aim of minimizing the least square errors in the reconstructed signal, PCA computes the total covariance structure from a given set of training data. The compact PCA subspace is found through eigen decomposition on the total covariance matrix. By projecting the feature data onto the PCA subspace, a number of uncorrelated features will be obtained, which can be best used for applications such as signal reconstruction and data compression. LDA, on another hand, is aimed for discrimination purposes. With a set of labeled training feature data from different classes, LDA computes the within-class covariance matrix and the between-class covariance matrix. A set of linear discriminant axes are found with the goal of maximizing the ratio between the between-class projected variance and the within-class projected variance. Practically, LDA suffers several problems: 1) The subspace found by LDA are prone to estimation errors on the covariance matrices, especially when numerous close-to-zeros eigen values of the within-class covariance matrix are present; 2) The number of discriminant features is limited by the number of classes. For an L-class classification problem, only L-1 discriminant features can be extracted by LDA; 3) the discriminant features found by LDA are statistically correlated. A large number of works are available in pattern classification literature to improve the subspace-based discriminant feature extraction through addressing the above issues of the LDA. In our work, we propose to use an eigenfeature regularization and extraction [114] technique for reliable extraction a compact set of discriminant forensics features. More details of this technique are explained in our Chapter 3 on mobile camera identification.

Unlike the subspace methods, feature selection techniques aim to select the most competent feature subset by removing the features identified to be unproductive or less productive towards a specific classification goal, e.g. maximizing the accuracy of a given classifier. Since the exhaustive searching of the best feature subset usually requires highly-intensive computation, the main issue becomes how to search for a best feature subset with manageable searching complexity. Pudil *et al.* [115] proposed a popular sequential forward floating search (SFFS) method together with nonmonotonic criterion function for feature selection. Starting from an empty feature set, the SFFS

algorithm performs stepwise feature inclusion and conditional feature removal to select a feature subset. Compared with simple sequential searching algorithms, the backtracking mechanism, i.e. the conditional feature removal, employed by SFFS allows removing the previous selected but currently useless features due to inclusion of new features, which more likely results in an optimal feature subset. The SFFS algorithm has been applied to several prior image forensics works [36, 49, 77, 116]. In this thesis, we have also used this technique to select a good subset of our demosaicing features in Chapter 2.

Classification Techniques

Among the different pattern classification techniques, support vector machine (SVM) is so far the most popularly used in the existing passive image forensics works. SVM is a powerful nonlinear pattern classification tool. By minimizing the structural risk [117] in classification, SVM optimizes the separation margin between different classes so that it typically exhibits good generalization performance on the unseen data. The kernel trick commonly employed in SVM also maps the current low-dimensional feature space into a high-dimensional space before a better separation plane can be found in the high-dimensional space. Several works in [118, 119] further extended the traditional SVM into probabilistic SVM (or PSVM) by studying distribution of the SVM outputs and maps them into probabilistic scores by optimizing a sigmoid function. In our work, we employ PSVM with the common radial basis function (RBF) kernel as a general tool for forensics classification and comparison purposes from our Chapter 2 to Chapter 5.

Boosting is another promising domain of classification methods that can be applied to image forensics. Boosting methods typically involves training a large number of so-called weak classifiers and incrementally combining them into a strong ensemble classifier. In the adaptive boosting or AdaBoost method [120], Freund and Schapire developed a theoretical framework to construct a strong ensemble classifier by incrementally minimizing the upper bound of an exponential loss function. Therefore, the separation margin is systematically improved to result in good generalization performances of an AdaBoost-learned classifier. In [121], Schapire *et al.* further elaborated several extensions of AdaBoost including the extension of the discrete AdaBoost into real-valued RealBoost algorithm and the extension of AdaBoost to

multiclass pattern classification. On top of the RealBoost, Li *et al.* [122] proposed a FloatBoost algorithm to incorporate a floating-search based backtracking mechanism to exclude the unfavorable classifiers from the classifier ensemble. It is claimed in [122] that FloatBoost typically results in less individual classifiers in an ensemble with comparable classification performances to the RealBoost. The boosting classifiers usually perform fast in classification due to the simple ensemble structure, which allow the boosting classifier being used in real-time applications, e.g. online classifications. Moreover, boosting algorithms can be easily applied to asymmetric learning tasks, where huge differences on the sizes of different classes are present. The cascade learning structure together with AdaBoost proposed in [123] further improves the flexibility of the decision-making process in real-time asymmetrical learning tasks. In this thesis, we propose a new FusionBoost algorithm to iteratively combine a set of pre-trained PSVM classifiers into a strong ensemble forensics classifier.

Some other classification methods applied in prior forensics works also include neural network [33], linear discriminant classifier [64] and probabilistic classifiers based on correlation similarity measure [34, 50].

Fusion Techniques

Classifier fusion refers to the techniques of combining multiple classifiers or features for a given pattern classification task. In image forensics, such techniques are useful for combining diversified forensics experts or features for better forensics performances. Classifier fusion can be implemented at different levels, such as feature level, kernel level, score level and decision level. At feature level, different features are simply combined before they are used in classification. Kernel trick is a common technique in state-of-the-art pattern classification techniques, e.g. SVM and neural network, to enable nonlinear classification with improved performances. The commonly used kernels include linear kernel, polynomial kernel, RBF kernel and sigmoid kernel. Fusion at kernel level can be implemented to compute the kernel values associated with different feature sets separately and combine these kernel values together to form a new hybrid kernel function. Score-level fusion is appropriate for combining classifiers, which provide confidences scores, e.g. probabilistic scores. The scores are combined and the

classification decision is made based on the combined score. For decision-level fusion, each individual classifier makes its own classification decision and the decisions are combined to give the final classification decision. For fusion at score level and decision level, fusion rules are commonly applied and the frequently used rules include the Mean, Summation, Product, Max, Min, Median and Majority Vote rules, etc. In a prior forensics work [49], Celiktutan *et al.* compare score-level fusion based on different rules with feature-level fusion in combining 3 diversified features for blindly identifying cellular phone camera models. Their comparison results show that the score-level fusion based on product rule gives the best result. It should be noted that the frequently-used fusion rules above do not study the output characteristics of the individual classifiers; hence the fusion performance is generally not optimized. Therefore, also as mentioned earlier, we have proposed a novel boosting-based fusion algorithm, called FusionBoost, to learn the characteristics of the individual probabilistic classifiers and to determine the weights for linearly combining the individual classifiers. The details of our proposed FusionBoost can be found in our Chapter 4 on universal image tampering detection.

1.3.6 Tamper Hiding Attacks

It should be noted that the forensics techniques are also subject to various attacks. The recent works in [124-126] have proposed several tamper hiding techniques, each targeted for a well-known passive forensics methodology. These techniques demonstrated that image forensics through detection of the resampling artifact in [76], the PRNU noise patterns in [34] and the color-filter-array interpolation traces in [75] can all be defeated. Though these attacks are specific to the selected forensics methodologies and each attack likely introduces new tampering artifacts detectable by other forensics methodologies, the works [124-126] have shown that performing image forensics with only a single tool can be easily attacked by a sophisticated attacker. It has also been pointed out in several early forensics works [67, 109] that comprehensive forensics analysis should be based on a suite of forensics tools which examine different image properties. In general, covering up all tampering artifacts and restoring all image regularities simultaneously into a tampered photo is believed to be very difficult than attacking on a single forensics tool. In the future, there is no doubt that the technology to doctor photos, including those for tamper hiding, will continuously improve and so will

be the image forensics techniques. It is hard to assure that the forensics side eventually will be the sole winner. However, we support Farid's opinion in [127] that "forensics would make it increasingly harder and more time-consuming to create compelling fake." By making image forgery a difficult task, the research effort in passive image forensics will still pay off to restore the traditional trustworthiness on photos especially in the occasions where image authenticity is important. This is analogous to the old time that photos were widely trusted simply because photo forgery via the darkroom tricks is very difficult.

1.4 Major Contributions and Organization

As mentioned earlier, the ultimate goal of this research is to restore the traditional trustworthiness on digital photos. In this thesis, this is done through improving reliability of the forensics detection and the decision making processes, especially for the forgeries that require little effort to create. The problems we address here are identification of various popular image source models, discriminative features extraction and fusion, universal image tampering discovery and prevention of the image recapturing threat. We describe a set of new forensics tools, whose efficacies are verified in relatively large-scale experimental tests. The proposed tools work in complete absence of pre-computed image hash or prior inserted information. Since image forgery is commonly created to fool human eyes, it is generally difficult for people to suspect on a delicately-made fake photo, to spot the tampered location and to make reliable final forensics decisions. However, the image manipulator frequently modifies some underlying image statistics, which are not directly visible to human eyes. Through detecting the anomalies and inconsistencies in these statistics and improving the decision-making process, our developed tools can be used either in automatic batch processing mode for daily routine-check on photos' authenticity or as facilitating tools to help the human forensics expert to better visualize the underlying statistical evidences. From Chapter 2 to Chapter 5, we describe in details our proposed tools for different forensics problems.

1. **Accurate detection framework of demosaicing regularity**: In Chapter 2, we propose an accurate detection framework to characterize the underlying demosaicing regularity introduced by different demosaicing algorithms. Through using partial

second-order derivative correlation models, our framework detects both the intra-color channel and cross-channel correlation. A two-level reverse classification scheme is also proposed to efficiently partition the color samples demosaiced with the same formula into the same demosaicing category. Both schemes reduce the detection variations caused by different image scenery and enable accurate estimation of the demosaicing parameters. Correspondingly, the extracted features show superior performances in identification of different image source models including 16 diversified demosaicing algorithms, 14 DSCs of different models and 10 commercial RAW tools over traditional demosaicing detection methods;

2. **Mobile camera model identification using eigenfeature regularization and extraction:** High dimensionality of statistical features and limited number of training images are a common problem faced in statistical image forensics as well as many other pattern recognition tasks. In Chapter 3, we propose to use our accurately detected demosaicing features together an eigenfeature regularization and extraction technique to extract a compact set of eigen demosaicing features for mobile camera model identification. Through comparison, we show that our eigen demosaicing features work extremely well and perform better than state-of-the-art forensics features in identifying 9 mobile cameras of dissimilar model labels. By further including a number of cameras of the same model or very close models, we also show that in a fifteen-cam identification experiment, our identification accuracies tend to mix among the very similar cameras to some extent. This justifies that our demosaicing features are more appropriate for camera model identification instead of individual camera identification. We have also tested using the eigen demosaicing features to identify the fourteen DSCs and the ten RAW tools in Chapter 2. We achieved slightly better identification accuracies with significantly less number of features than the SFFS feature selection technique we employed in Chapter 2.

3. **Ensemble tampering detection using FusionBoost**: Given a suite of probabilistic forensics experts, how can the diversified probabilistic scores be combined to give a better final decision? In Chapter 4, we propose an ensemble tampering detection framework using FusionBoost and the accurately detected demosaicing features in Chapter 2. By first training a set of individual probabilistic tampering detectors using different demosaicing feature sets, we linearly combine the individual

classifiers where the classifier weights are determined through a proposed iterative learning procedure called FusionBoost. In this framework, we address the issue of large asymmetry in the tampering detection task with suitable solutions provided.

4. **Identification of recaptured image on LCD screens**: Image recapturing is an easy way to hide many tampering artifacts and to restore the various image regularities. Such processes potentially defeat many current forensics systems and at the same time, deceive human eyes. In Chapter 5, we study the best setting to recapture good-quality photos on the ubiquitous LCD screens in a realistic environment and survey on the human beings' ability to identify these recaptured photos. By observing the artifacts introduced by the image recapturing process, we also propose using several types of image features to identify the recaptured photos on LCD screens from the natural photos.

Lastly in Chapter 6, we draw the conclusions and suggest some future directions.

Chapter 2 Accurate Detection of Demosaicing Regularity for Image Source Model Identification

Color filtering and demosaicing are common processes in a commercial single-sensor camera to cost-effectively produce color. Many different demosaicing algorithms are present in the literature and camera manufacturers and RAW-tool makers employ their proprietary demosaicing techniques. These different demosaicing algorithms likely introduce distinctive and consistent pixel correlation in the reconstructed three-color images, which can be detected for forensics purposes. In this chapter, we propose an accurate detection framework of demosaicing regularity from different source images. The theoretic foundation and the important procedures are explained in details. Experimentally, we show effectiveness of the proposed method in estimating the underlying demosaicing formulas and in identifying various image sources through comparing with previous demosaicing detection methods.

2.1 Commercial Digital Still Cameras

Depending on the number of sensor arrays used, commercial digital still cameras are broadly classified into two categories, the single-sensor based and the three-sensor based as illustrated in Fig. 2-1. The three-sensor camera in Fig. 2-1(b) has three sensor chips, each for capturing a specific color. A beam splitting prism is often required to direct

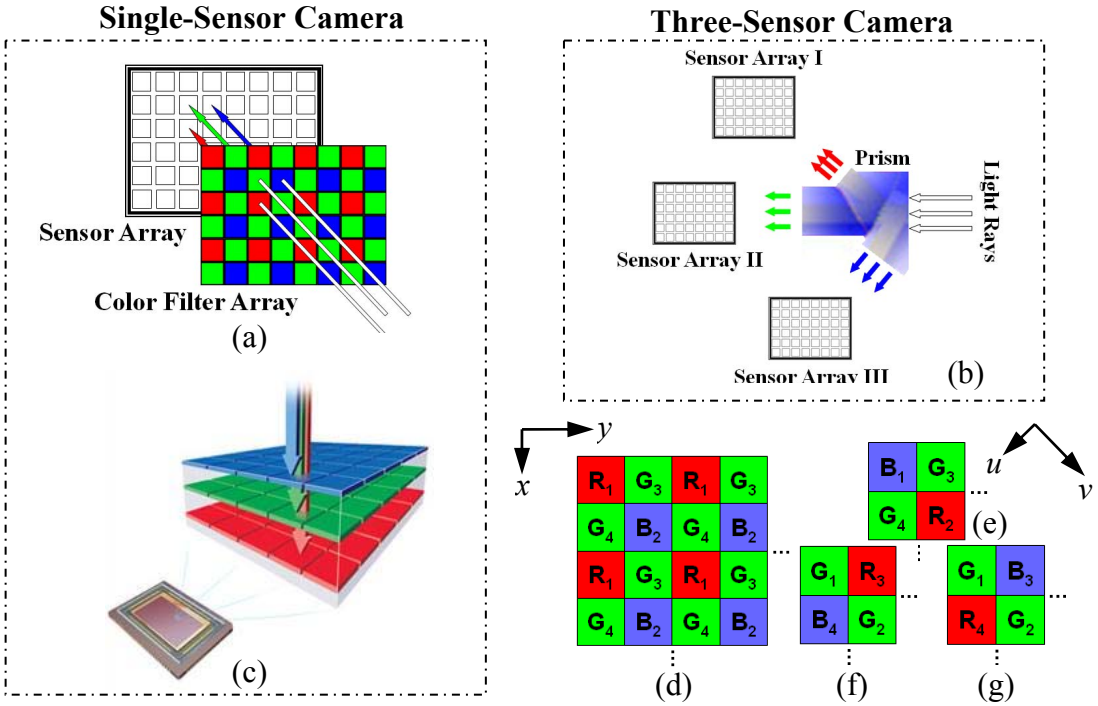


Fig. 2-1 Single-Sensor Camera System and Three-Sensor Camera System; (a) Illustration of a Single-Sensor System; (b) Illustration of a Three-Sensor System; (c) Foreon X3 Image Sensor [128]; (d) Bayer CFA Pattern [129] and (e),(f),(g) the Three Shifted Bayer CFAs

light photons of different wavelengths (colors) to its appropriate sensor arrays. A three-sensor camera usually produces better color fidelity since all the three color components are captured at each pixel location. However, using three sensor arrays is very costly and often makes the camera system clumsy. Such design is only used in certain high-end cameras and the commercially available DSCs are mostly based on a single sensor array. Since the common CCD and CMOS image sensors can hardly differentiate colors, majority of the single-sensor cameras use color filtering in conjunction with demosaicing to cost-effectively produce color. The color filter array (CFA) are mounted on top of the sensor array as shown in Fig. 2-1 (a), allowing only one specific color being captured at each pixel location. The different types of color filters are typically arranged in 2×2 periodical lattices and Bayer CFA patterns in Fig. 2-1(d) are the dominantly used CFA pattern [144] in the commercial DSCs. Since a full-color image requires three color components at each pixel, the missing two colors at each pixel location are later reconstructed from their neighboring sensor samples through an interpolation process, commonly known as demosaicing.

A notable exception to the common image sensors is the Foveon X3 CMOS sensor, which simultaneously captures three different colors at each sensor location (pixel) [128]. Announced by Foveon Inc. in 2002, this CMOS sensor is fabricated with three embedded layers in silicon to exploit the fact that different wavelengths (colors) of light penetrate silicone into different depths. Currently, this special sensor is still limited to a few camera models including several Sigma camera models, a Toshiba Teli microscopy camera and a FoMOS camera development platform. Hardly any of these models are highly rated to represent the mainstream in today's commercial DSC technology according to the digital photography review in [130].

2.2 Demosaicing and Existing Detection Methods

Majority of existing photos are still captured by the single-sensor DSCs, where the color is produced through color filtering and demosaicing techniques. As a key process that determines fidelity of a color photo, demosaicing has been extensively studied [129, 131-144] and the camera manufactures typically implement their proprietary demosaicing techniques. Below we summarize the major differences related with the color filtering and demosaicing processes.

- *CFA pattern*: The most commonly used is Bayer CFA pattern in Fig. 2-1(d-g) [129] though other CFA patterns, such as Fujifilm's SuperCCD and complimentary CFA patterns like CMY and CMYG, are also available;
- *Grouping*: The missing color samples are typically grouped before an appropriate demosaicing formula is applied for each group. The grouping can be edge-adaptive or non-edge-adaptive. For edge-adaptive algorithms, the decision criteria also differ significantly for the distribution of the samples to different edge groups;
- *Reconstructive filtering*: The reconstruction is commonly performed in the color-difference domain. Other possible domains include the intra-channel domain and the color-hue domain. The low-pass reconstruction filters can differ significantly in their kernel parameters and sizes;
- *Refinement and enhancement*: As an optional step, the refinement is commonly iterated in updating the green channel and then the red and blue channels. The updating formulas also differ in their parameters.

Several previous forensics works have characterized the demosaicing regularity for both image source identification and tampering detection. The work in [27, 75] used an expectation maximization (EM) technique to compute a set of weights for classification of several demosaicing algorithms and for identification of camera models. The works in [28] extended the EM technique by using both the derived EM weights and the average second-order derivative spectrum as features for camera identification. In [33], quadratic pixel correlation coefficients are proposed as demosaicing features for camera model identification. Though reasonable accuracies for identification of three to four commercial camera models are reported, these methods are not practical due to an overly simplified implicit assumption, i.e. each pixel is equally correlated with its neighboring pixels in a color channel. The work in [45] introduces several new concepts:

1. Detection of the CFA pattern;
2. Heuristic division of the image into three regions, the horizontal edges, the vertical edges and the smooth region with an implicit assumption that the demosaicing formula for each region is similar;
3. Each demosaiced sample is written as a weighted average of its neighboring sensor samples in the same color channel and the optimal weights are solved as a total least square solution [145].

Inspired by the new concepts, we also note that this detection method has two major drawbacks:

1. The incapability of capturing the cross-color channel correlations caused by demosaicing;
2. The heuristic division is rough, which largely depends on an empirical threshold and cannot accurately reveal the true varying demosaicing grouping for diversified demosaicing algorithms.

In this Chapter, we consider accurate detection of the image demosaicing regularity. This is motivated by the fact that 2/3 of the color samples of a common photo are reconstructed by a demosaicing algorithm consisting of only a few formulas and the large population of demosaiced samples provides a good basis for reliable statistical characterization of the applied demosaicing technique. With an assumption that various

demosaicing algorithms reconstruct smooth samples from its neighboring sensor samples, we compute in Section 2.3 the partial second-order derivative of a demosaiced sample. This derivative provides rich information of the applied demosaicing formula and at the same time, it does not contain the local DC component within a color channel. By proposing a partial derivative correlation model, our estimation of the underlying demosaicing formulas is naturally extended across the boundaries of color channels. In addition, a proposed reverse classification scheme in Section 2.4 precisely classifies samples demosaiced by the same formula into the same category, which minimizes the detection variations due to the reverse classification errors. Both the partial derivative correlation model and the reverse classification improve the detection accuracy by suppressing content-dependant estimation variations for color images undergone the same demosaicing process. Three types of features computed from 16 categories of demosaiced samples comprehensively represent the regularity introduced by demosaicing. Since forensics challenges are commonly formulated as pattern classification problems, by enhancing the detection accuracy and by improving comprehensiveness of the feature description, we suppress the within-class feature variations and enlarge the between-class separation. Consequently, this leads to the superior forensics performances especially when our proposed method is compared with several existing demosaicing detection methods in Section 2.5.

2.3 Partial Derivative Correlation Model

2.3.1 Image Derivative on 1D Periodical Lattice

Since common CFAs reside on a 2×2 periodical lattice as in Fig 2.1(d) and demosaicing is frequently carried out along one axis, we first derive the second-order derivative formula based on a 1D periodical mosaic lattice in Fig. 2-2, where $\{F(n), n=1, 2,\ldots\}$ can be viewed as discrete samples from a smooth continuous function $f(t)$ at equal sampling intervals. The indexes $\{\ldots, q-1, q+1, \ldots\}$ and $\{\ldots, q-2, q, q+2, \ldots\}$ are respectively associated with the sensor samples and the demosaiced samples.

Suppose $F(q)$ is demosaiced along the t-axis, it is reasonable to assume $f(t)$ is

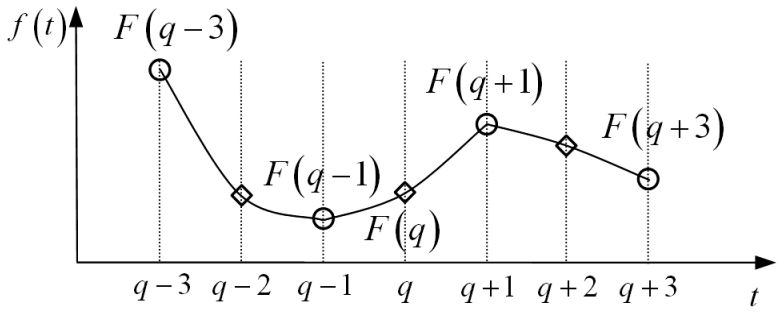


Fig. 2-2 A One Dimensional Array of Periodical Color Samples Extracted from One Color Channel of a Demosaiced Image, Where 'o's are Sensor Samples and '◊'s are Demosaiced Samples. *F(q)* is Demosaiced Along *t*-Axis

smooth at $t=q$ such that both the left and the right derivatives of $f(t)$ are continuous at $t=q$, and Taylor series expansions are applicable below.

$$F(q+d)=F(q)+\frac{f^{1}(q)}{1!}d+\frac{f^{2}(q)}{2!}d^{2}+\frac{f^{3}(q)}{3!}d^{3}+\cdots \tag{2.1}$$

With the second-order approximation on the Taylor series expansions, we consider below a small neighborhood including the nearest sensor samples $F(q\text{-}1)$ and $F(q\text{+}1)$,

$$\begin{cases} F(q+1)=F(q)+f^{(1)}(q)+f^{(2)}(q)/2 \\ F(q-1)=F(q)-f^{(1)}(q)+f^{(2)}(q)/2 \end{cases} \tag{2.2}$$

From Eqn (2.2),

$$f^{(2)}(q)=F(q-1)+F(q+1)-2F(q) \tag{2.3}$$

Or equivalently,

$$F(q)=\boldsymbol{\alpha}^{T}\boldsymbol{\gamma}-\beta f^{(2)}(q) \tag{2.4}$$

where $\boldsymbol{\alpha}=[0.5\ 0.5]^{T}$, $\boldsymbol{\gamma}=[F(q\text{-}1)\quad F(q\text{+}1)]^{T}$ and $\beta=0.5$. Since both $F(q\text{-}1)$ and $F(q\text{+}1)$ are known sensor samples, $F(q)$ is linearly correlated with and directly computable from $f^{(2)}(q)$, its second-order derivative along the t-axis. Therefore, various demosaicing

algorithms that estimate $F(q)$ along the t-axis are equivalent to estimate $f^{(2)}(q)$ first, followed by applying a known linear transformation in Eqn (2.4).

The above formulation can be extended to a higher-order approximation by considering a larger neighborhood of sensor samples. For instance, with the fourth-order approximation, we can further include sensor samples $F(q\text{-}3)$ and $F(q\text{+}3)$ to form a set of equations in a similar manner to those in Eqn (2.2). Through elimination, similar second-order derivative formula to Eqn (2.3) can be derived. In such a case, Eqn (2.4) still holds with α, γ and β changed correspondingly. One can refer to the Appendix A for our derivation of the second-order derivative formula based on fourth-order and sixth-order approximation in the Taylor series expansion.

Since only second-order derivatives are used in our formulation, in what follows in this chapter, we use 'derivative' to represent 'second-order derivative'.

2.3.2 Derivative-Based Demosaicing Model

To detect the image regularity associated with demosaicing, it is important to have a generalized model to represent the demosaicing processes so that different demosaicing formulas differ only in some free model parameters. Below, we propose a partial derivative correlation model for such a purpose.

To facilitate the discussion, we let $\mathbf{D}=\{D_{ijc}\}$ represent a 3D array of the demosaiced image of size $H\times W\times K$, where H and W are the height and width of the image, respectively, and K=3 denotes the number of color channels. Symbol c is to indicate the red, green and blue channels for a Bayer CFA. Suppose the CFA pattern is known, we can sample the 2D RAW image $\mathbf{S}=\{S_{ij}\}$ from $\mathbf{D}$ accordingly. As an example, $\mathbf{D}$ and $\mathbf{S}$ for the first Bayer CFA pattern in Fig. 2-1(d) can be represented as

$$\mathbf{D}=\{D_{ijc}\}=\begin{pmatrix}\{r,G,B\}_{11} & \{R,g,B\}_{12} & \cdots\\ \{R,g,B\}_{21} & \{R,G,b\}_{ij} & \\ \vdots & & \ddots\end{pmatrix},\quad \mathbf{S}=\{S_{ij}\}=\begin{pmatrix}r_{11} & g_{12} & \cdots\\ g_{21} & b_{ij} & \\ \vdots & & \ddots\end{pmatrix} \tag{2.5}$$

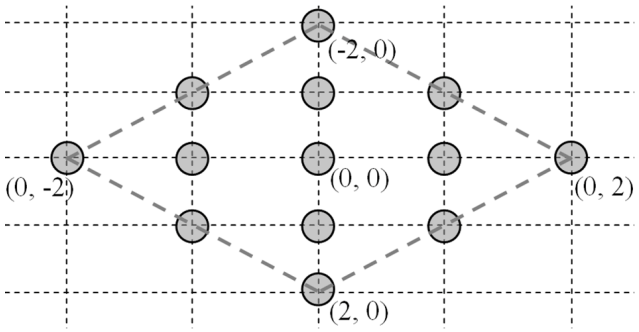


Fig. 2-3 Diamond Weight Pattern

where capital letters *R*, *G* and *B* denote the demosaiced samples in the red, green and blue color channels, respectively, and lower-case letters *r*, *g* and *b* denote the sensor samples.

Suppose a sample D_{ijc} is demosaiced along *t*-axis, where $t \in \{x,y,u,v\}$ as depicted in Fig. 2-1(d) and since from Eqn (2.4), demosaicing D_{ijc} along *t*-axis is equivalent to estimating its partial derivative $D_{ijc}^{(t)}$ along the *t*-axis, we can write a general demosaicing equation below

$$D_{ijc}^{(t)} = \sum_{\forall (p,q) \in \Omega} w_{pq}^{(t)} S_{i+p,j+q}^{(t)} + e_{ijc}^{(t)} \tag{2.6}$$

where $S_{ij}^{(t)}$ denotes a supporting RAW partial derivative computed from **S** along the *t*-axis, $\{w_{pq}^{(t)}\}$ are weights of the supporting derivatives $\{S_{ij}^{(t)}\}$, which represent the formula used to demosaic D_{ijc}, $e_{ijc}^{(t)}$ is the corresponding estimation error and $\Omega=\{(p, q)$, where $p^2 + q^2 \le 4\}$ defines a diamond weight pattern as shown in Fig. 2-3. Based on the second-order approximation, we use the derived derivative equation in Eqn (2.3) to compute the partial derivatives. For instance, the *x*-axis derivatives are

$$\begin{aligned} D_{ijc}^{(x)} &= D_{i-1,jc} + D_{i+1,jc} - 2D_{ijc} \\ S_{ij}^{(x)} &= S_{i-2,j} + S_{i+2,j} - 2S_{ij} \end{aligned} \tag{2.7}$$

Note that **S** contains sensor samples of all three color channels and the samples $S_{i-2,j}$, S_{ij} and $S_{i+2,j}$ belong to the same color channel due to the 2×2 periodicity of the commonly

used CFAs, e.g. the Bayer CFAs in Fig. 2-1(d)-(g). This CFA periodicity also applies for other axes including *y*, *u* and *v*. The partial derivative formulas along the *y*- , *u*- and *v*-axes can be written similarly to Eqn (2.7).

We also note that for a number of demosaicing algorithms, one color plane *l* with rich luminance information (green for a Bayer CFA) is reconstructed first. The reconstruction of another color plane *h* with rich chrominance information (red or blue for a Bayer CFA), is based on both the fully-populated color plane *l* and the RAW **S**. To cater for such cases, we modify the demosaicing equation in Eqn (2.6) accordingly for a demosaiced sample D_{ijh} in the *h* color plane as

$$D_{ijh}^{(t)} = \sum_{\forall(p,q)\in\Omega} w_{pq}^{(t)} S_{i+p,j+q}^{(t)} + \sum_{\forall(p,q)\in\Omega} \mu_{pq}^{(t)} D_{i+p,j+q,l}^{(t)} + e_{ijh}^{(t)} \tag{2.8}$$

where $\{\mu_{pq}^{(t)}\}$ are weights of the supporting derivatives $\{D_{ijl}^{(t)}\}$ along *t*-axis.

Since the supporting derivatives $\{S_{ij}^{(t)}\}$ are computed from the sensor samples of all three color channels, our correlation models in Eqn (2.6) and Eqn (2.8) simultaneously take into account both the cross- and the intra-channel correlation. This is important as the state-of-the-art demosaicing algorithms often employ the color difference or the hue domains for demosaicing and this inevitably introduces strong cross-channel correlation. The main advantage of using the partial derivative correlation models is that these derivatives do not contain any local DC component of the respective color channel, which allows estimation of the underlying demosaicing weights, being naturally extended across the boundary of different color channels. Comparatively, the conventional pixel-based correlation model can be hardly extended across the boundary of color channels as the local DC levels in all three color channels are highly scenery dependant and generally unequal. By removing the local DC components, the derivative-based formulation focus on the image correlation caused by demosaicing and the estimation variations caused by different image sceneries are largely suppressed.

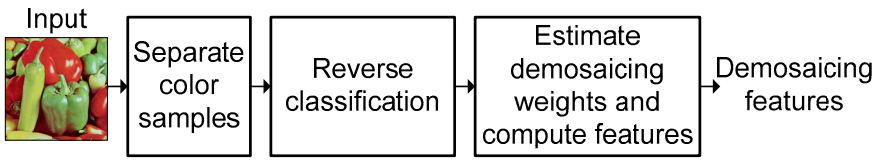


Fig. 2-4 Overview of the Proposed Detection Framework

2.4 Proposed Detection Framework

Shown in Fig. 2-4 is an overview of the proposed detection framework based on the partial derivative correlation model. Since Bayer CFA has been dominantly used in commercial DSCs [144], we first separate the sensor samples from demosaiced samples according to a Bayer CFA. Then a reverse classification scheme exclusively partitions all demosaiced samples into sixteen categories. An expectation maximization reverse classification (EMRC) algorithm is employed to resolve the ambiguous demosaicing axes. After the sixteen categories are formed, the partial derivative correlation models are used to form a set of linear demosaicing equations and the weights are estimated as a regularized least square solution. Three types of demosaicing features are computed from the sixteen categories for image forensics applications. In the following section, we elaborate the steps in details.

2.4.1 Reverse Classification to Estimate Demosaicing Weights

After separating the sensor samples **S** and identifying the demosaiced samples from a given demosaiced image **D** according to a Bayer CFA, we perform reverse classification to exclusively classify all demosaiced samples into a number of categories so that each category of demosaiced samples is reconstructed by the same or very similar formulas. The goal of the reverse classification is therefore to best recover the implicit grouping adopted by the underlying demosaicing algorithm. Precise reverse classification is a prerequisite to enable accurate estimation of the demosaicing formulas.

As shown in Fig. 2-5, our proposed reverse classification is implemented in two levels. In the first level, we divide the demosaiced samples according to their color channel and the relative positions in the 2×2 periodical CFA lattice. As a result, we form

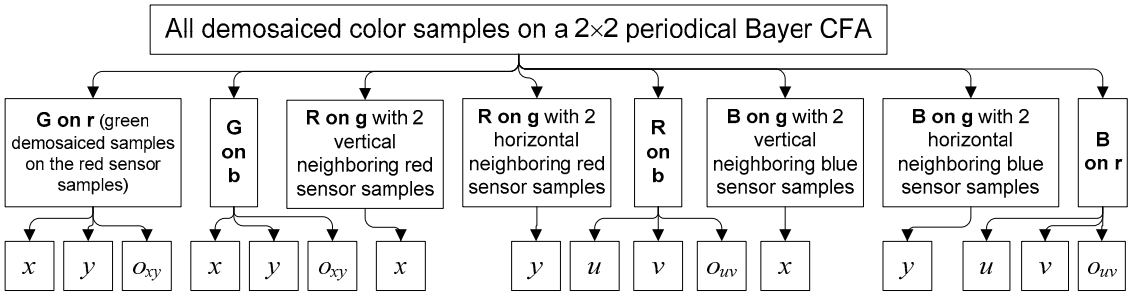


Fig. 2-5 Two-Level Reverse Classification of the Demosaiced Samples for Bayer CFA into 16 Categories with the Demosaicing Axes Indicated

eight first-level categories including two green categories and three categories each from the red and blue channels. In the second level, we first determine the possible demosaicing axes for each first-level category and further partitions the first-level categories according to their possible demosaicing axes. As the first-level reverse classification is straightforward, below we focus on the second level.

For each of the eight first-level categories, the possible demosaicing axes are determined by examining pattern of nearest surrounding sensor samples in the same color channel. Fig. 2-6 shows four possible patterns for Bayer CFA. For patterns (a) and (b) in this figure, ambiguity arises as demosaicing can be conveniently carried out along either one of the two suggested axes or omnidirectionally, i.e. as an average of both axes. To resolve this ambiguity, we further partition such a first-level category into three second-level categories according to the three demosaicing axes by using an expectation maximization reverse classification (EMRC) algorithm below. The goal of this EMRC algorithm is to jointly assign each sample to its most appropriate demosaicing axes and at the same time, to estimate the corresponding demosaicing formulas.

We first consider the green-channel estimation and let $\{z_n,\ 1 \le n \le N\}$ denote a first-level green category, where $N = \lfloor H \times W / 4 \rfloor$. The demosaiced samples $\{z_n\}$ are re-organized based on a single index n. This category is associated with the pattern in Fig. 2-6(a) and the possible demosaicing axes are x, y and o_{xy}, i.e. average of the x- and y-axes. Let $\{z_n^{(1)}\}$, $\{z_n^{(2)}\}$ and $\{z_n^{(3)} = (z_n^{(1)} + z_n^{(2)})/2\}$ denote the corresponding x-, y-axes derivatives and the average derivatives respectively. For each sample z_n and each axis $t \in \{1, 2, 3\}$, we follow Eqn (2.6) to get

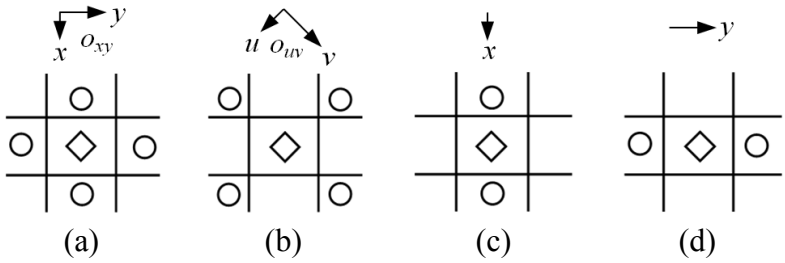


Fig. 2-6 Four Patterns of Nearest Sensor Samples for Bayer CFA, ("o": Sensor Samples; "◊": Demosaiced Samples); For Green Channel (a) Four Direct Neighbors; and For Red and Blue Channels (b) Four Corner Neighbors; (c) Two Vertical Neighbors and (d) Two Horizontal Neighbors.

$$e_n^{(t)} = z_n^{(t)} - \mathbf{w}^{(t)T}\mathbf{s}_n^{(t)} \tag{2.9}$$

where $\mathbf{w}^{(t)}$ is a vector of the weights for the t^{th} axis, $\mathbf{s}_n^{(t)}$ is a vector of the corresponding supporting RAW derivatives and $e_n^{(t)}$ denotes the estimation error. The vectors $\mathbf{w}^{(t)}$ and $\mathbf{s}_n^{(t)}$ are formed by re-aligning all elements in the diamond weight pattern defined in Fig. 2-3 row by row into single-column vectors. For the o_{xy}-axis, $\mathbf{s}_n^{(3)}$ contains both the *x*- and *y*-axes supporting derivatives.

Since the demosaicing weights $\mathbf{w}^{(t)}$ are unknown, directly assigning the most appropriate demosaicing axis for each sample z_n is impossible. Therefore, the proposed EMRC algorithm iteratively minimizes

$$J = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{3} v_n^{(t)}\left(z_n^{(t)} - \mathbf{w}^{(t)T}\mathbf{s}_n^{(t)}\right)} \tag{2.10}$$

where $v_n^{(t)} \in \{0,1\}$ is the assignment indicator and $v_n^{(t)} = 1$ indicates that z_n is assigned to the t^{th}-axis.

The weights for $\mathbf{w}^{(t)}$ are initially set to zero. In the **expectation step**, we update

$$v_n^{(t)} = \begin{cases} 1, \text{ if } t = \arg\min\left(\left|z_n^{(t)} - \mathbf{s}_n^{(t)T}\mathbf{w}^{(t)}\right|\right) \\ 0, \text{ otherwise} \end{cases} \tag{2.11}$$

for n=1,…, N while $\mathbf{w}^{(t)}$ is fixed.

In our proposed **maximization step**, we fix $v_n^{(t)}$ and compute $\mathbf{w}^{(t)}$ by minimizing the following criterion

$$E\left(\mathbf{w}^{(t)}\right) = \left\|\mathbf{Q}^{(t)}\mathbf{w}^{(t)} - \mathbf{z}^{(t)}\right\|^2 + \lambda\left\|\mathbf{w}^{(t)}\right\|^2 \tag{2.12}$$

By differentiating $E\left(\mathbf{w}^{(t)}\right)$ with respect to $\mathbf{w}^{(t)}$, we derive

$$\frac{\partial E\left(\mathbf{w}^{(t)}\right)}{\partial \mathbf{w}^{(t)}} = 2\left(\mathbf{Q}^{(t)T}\mathbf{Q}^{(t)} + \lambda\mathbf{I}\right)\mathbf{w}^{(t)} - 2\mathbf{Q}^{(t)T}\mathbf{z}^{(t)} \tag{2.13}$$

By setting the results to zero and rearranging the terms, we arrive at the regularized least square solution [138],

$$\mathbf{w}^{(t)} = \mathbf{Q}^{(t)\dagger}\mathbf{z}^{(t)} \tag{2.14}$$

for t = 1, 2 and 3, where

$$\mathbf{Q}^{(t)} = \begin{bmatrix} v_1^{(t)}\mathbf{s}_1^{(t)T} \\ \vdots \\ v_N^{(t)}\mathbf{s}_N^{(t)T} \end{bmatrix}, \mathbf{z}^{(t)} = \begin{bmatrix} v_1^{(t)}z_1^{(t)} \\ \vdots \\ v_N^{(t)}z_N^{(t)} \end{bmatrix}, \ \mathbf{Q}^{(t)\dagger} = \left(\mathbf{Q}^{(t)T}\mathbf{Q}^{(t)} + \lambda\mathbf{I}\right)^{-1}\mathbf{Q}^{(t)T} \ ,$$

$\|\cdot\|$ denotes Frobenius norm, λ is a small regularization constant which prevents overfitting and improves stability of the solutions especially under ill conditions. **I** denotes an identity matrix. Since $\mathbf{w}^{(t)} = \mathbf{0}$ is also a valid solution but Eqn (2.14) hardly returns such a special solution, we set $\mathbf{w}^{(t)} = \mathbf{0}$ if $\left\|\mathbf{z}^{(t)}\right\| \le \left\|\mathbf{Q}^{(t)}\mathbf{w}^{(t)} - \mathbf{z}^{(t)}\right\|$.

The above expectation and maximization steps are repeated until the following stabilization condition is met,

$$J^{(i-1)} - J^{(i)} < T_h \tag{2.15}$$

where i denotes the current iteration number and T_h is a small experimentally determined constant. Both the expectation and maximization steps are designed to reduce J, which is sharply reduced at the few initial iterations and gradually stabilized to a low level. T_h is chosen to speed up the iteration process for the EMRC algorithm. By setting T_h to be close to zero, more iterations are needed. Experimentally, with a small T_h in the range of [0.001 0.005], the required number of iterations drops significantly. As the outcome of the EMRC algorithm, the true grouping employed by the underlying demosaicing technique can be largely recovered and the weights $\mathbf{w}^{(t)}$ representing the underlying demosaicing formulas are more reliably estimated.

For the red and blue ambiguous first-level categories associated with pattern (b) in Fig. 2-6, the reverse classification is implemented in a similar manner to partition the samples according to the u, v and o_{uv} axes. The main difference is that the demosaicing equations represented similarly to Eqn (2.9) are written by following the derivative correlation model in Eqn (2.8), where $\mathbf{s}_n^{(t)}$ contains the supporting derivatives computed from both the RAW image and the green plane.

This EMRC algorithm can be extended to the cases where a first-level category needs to be more thoroughly divided, i.e. when more than three sub-categories are needed. Such a scenario is necessary when the regularity of complex demosaicing algorithms needs to be thoroughly investigated.

For patterns (c) and (d) in Fig. 2-6, the demosaicing axes are clear as the nearest two sensor samples are either vertically or horizontally arranged. Such first-level categories are directly exported to the second level and the corresponding demosaicing weights are computed similarly to the maximization step of the above EMRC algorithm.

2.4.2 Computation of Demosaicing Features

With the optimal weights determined for the sixteen demosaicing categories, we compute three types of features below.

Weights (WT): These features represent the applied demosaicing formulas with a total of 312 weights computed for the 16 demosaicing categories.

Error Cumulants (EC): With the optimal weights **w** available, the absolute errors are given below

$$|\mathbf{e}| = |\mathbf{Qw} - \mathbf{z}| \tag{2.16}$$

where **Q** is the matrix of supporting derivatives and **z** is the vector of the corresponding derivatives. For each category, we compute four error cumulants including the mean, variance, skewness and kurtosis. For sixteen categories, a total of 64 error cumulants are computed, which statistically reveal the goodness that our estimated weights fit the underlying demosaicing algorithm.

Normalized Group Sizes (NGS): After partitioning an ambiguous first-level category using the EMRC algorithm, percentage of the demosaiced samples distributed to the 3 sub-categories are good indicators of the implicit grouping adopted by the underlying demosaicing algorithm. The normalized sizes of the sub-categories in percentages are also included as our features. For a total of four ambiguous first-level categories, we compute eight such features.

In commercial digital still cameras, all the four versions of Bayer CFAs in Fig. 2-1 (d)-(g) are likely adopted. Hence we apply our detection method four times to cater for the four possibilities. Consequently, we obtain 312×4=1248 weights, 64×4=256 error cumulants and 8×4=32 normalized group sizes. Though the feature dimension increases four-fold, some key advantages are: 1) detection of the correct Bayer CFA is not needed and the discriminative CFA information is automatically included in the overall feature set; 2) the correct Bayer CFA must be present if Bayer CFA is used. We find the features computed based on the three incorrect Bayer CFAs also provide useful relative information and the four channels of demosaicing features statistically form stable and unique patterns; 3) other non-Bayer CFAs, e.g. SuperCCD and complimentary-color CFAs such as CMY and CMYG, share common properties as Bayer CFAs including 2×2 periodicity and mosaic lattice, which make our derivative correlation models still applicable to such CFAs. Demosaicing based on these CFAs will also cause stable and unique imbalance pattern among these 4 channels of statistical features and our demosaicing features based on four Bayer CFAs help comprehensively represent this

Table 2-1 Comparison of Sixteen Conventional Demosaicing Algorithms

ID	Algorithm	Edge Adaptiveness	Domain (Refinement)	Reconstructive Filter (Size)
1	Bilinear [75]	Non-adaptive	Intra-channel	Fixed (3×3)
2	Bicubic [75]	Non-adaptive	Intra-channel	Fixed (7×7)
3	Cok 86 [131]	Adaptive	Intra-channel	Mixed (≥3×3)
4	Cok 87 [132]	Non-adaptive	Color hue	Fixed (3×3)
5	Freeman 88 [133]	Non-adaptive	Color diff.	Median (3×3)
6	Laroche 94 [134]	Adaptive	Mixed	Fixed (3×3)
7	Hamilton 97 [135]	Adaptive	Color diff.	Fixed (3×3)
8	Kimmel 99 [136]	Adaptive	Color hue (Inverse diffusion)	Adaptive (3×3)
9	Li 01 [137]	Non-adaptive	Color diff.	Adaptive
10	Gunturk 02 [138]	Non-adaptive	Color diff. (Iterative)	Fixed (5×5)
11	Pei 03 [139]	Adaptive	Color diff. (Iterative)	Adaptive (3×3)
12	Wu 04 [144]	Adaptive	Color diff. (Iterative)	Fixed (3×3)
13	Chang 04 [140]	Adaptive	Color diff. (Iterative)	Adaptive (5×5)
14	Wang 05 [143]	Adaptive	Color diff. (Iterative)	Adaptive (3×3)
15	Hirokawa 05 [142]	Adaptive	Color diff. (Iterative)	Median (5×5)
16	Alleysson 05 [141]	Non-adaptive	Chrominance	Fixed (11×11)

(a)

		0		
	0	0	0	
0	0	1	0	0
	0	0	0	
		0		

(b)

		-0.0040		
	0.0019	-0.0033	0.0021	
0.0020	-0.0020	0.9956	-0.0002	0.0015
	0.0033	-0.0045	0.0018	
		-0.0026		

(c)

		-0.0597		
	-0.0264	0.1689	-0.0376	
-0.1785	0.3371	0.4816	0.3468	-0.1682
	-0.0181	0.1292	-0.0344	
		-0.0673		

(d)

		-0.1076		
	0.1442	-0.1401	0.1121	
-0.2587	-0.1448	0.6071	-0.1627	-0.2717
	0.1216	-0.1142	0.1533	
		-0.0862		

(e)

		-0.1777		
	0.1997	-0.0822	0.1805	
-0.1269	-0.4602	0.8319	-0.4794	-0.1272
	0.1986	-0.0673	0.1783	
		-0.1663		

(f)

Fig. 2-7 Comparison of the Estimated Weights Based on Different Bayer CFAs for the **G on *r***, *x*-axis Category (Refer to Fig. 2-5 for This Category); (a) A Demosaiced Color Image (512×512) Using Hamilton's Algorithm [135], where Bayer CFA in Fig. 2-1 (d) is the Correct CFA; (b) Ground-Truth Derivative-Based Weights Mapped from Hamilton's Demosaicing Algorithm; (c) Estimated Weights According to the Correct Bayer CFA in Fig. 2-1 (d); (d), (e) and (f) are the Estimated Weights According to Shifted Bayer CFAs in Fig. 2-1 (e), (f) and (g) Respectively

statistical imbalance.

The proposed total 1536 features comprehensively represent the demosaicing regularity of all sixteen demosaicing categories. The overall feature computational time averaged over 100 runs on a P4-2.66 GHz PC with MATLAB 7.4 for 100 different color images of sizes 512×512 is about 45 seconds.

In view of the high dimensionality, which requires huge training data and causes slow classifier training and forensics response, we reduce the feature dimension by selecting a compact subset of features using sequential forward floating search (SFFS) algorithm [115]. Starting from an empty feature set, the SFFS algorithm performs stepwise feature inclusion and conditional feature removal to select a subset of features. Compared with simple sequential searching algorithms, the backtracking mechanism employed by SFFS allows removing the previous selected but currently useless features due to inclusion of new features, which more likely results in an optimal feature subset.

In the actual implementation, we find that SFFS feature selection would take impractically long time especially when our feature size is as large as 1536, the number of features to be selected is large, many training samples are involved and the criterion is based on optimizing a complex classifier. To make our SFFS time practically tractable, we perform the feature selection in two passes. In the first pass, we divide all 1536 demosaicing features into twenty subsets according to the demosaicing categories and feature types with an average feature size of about 77. We perform SFFS feature selection within each feature subset to select no more 30 features based on relatively compact training image sets. A score is then computed for each feature by measuring its accumulative contribution to the cross-validation accuracy within its feature subset. Around 450 features with leading positive scores are selected in the first pass. In the second pass, we pool the selected features in the first pass together and perform SFFS feature selection again to select a total of 300 features. As the outcomes of the second pass, we have 300 selected features in sequence. In the above SFFS process, the accuracy of a simple k-nearest neighbor (KNN) classifier with k=3 is used as the optimization criterion and the number of training samples used in each SFFS step is adjusted to be no more than 3000.

2.5 Simulation Result and Discussion

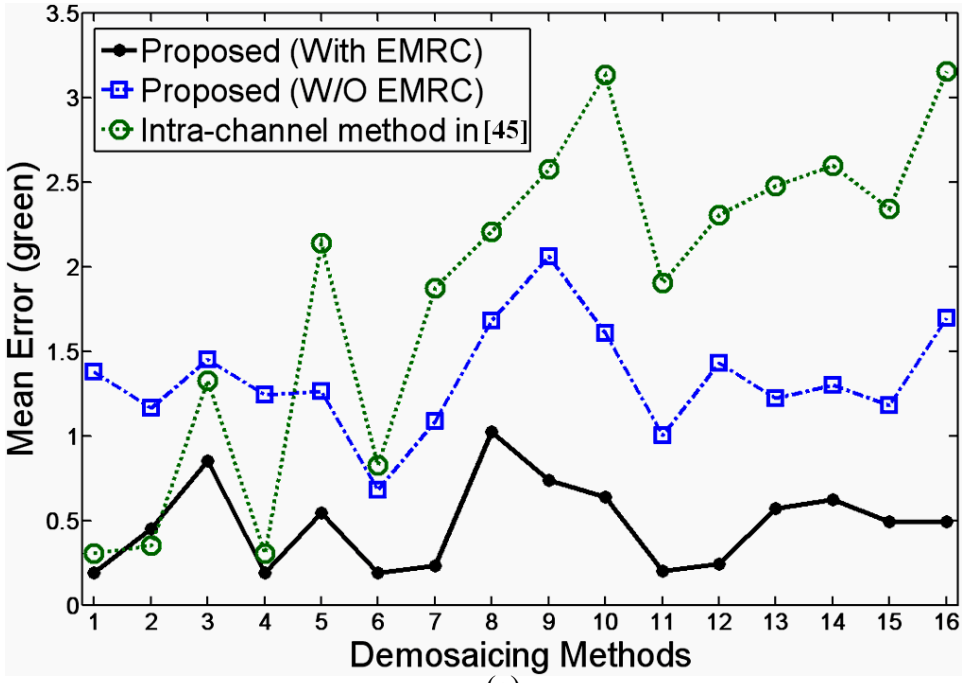
In our experiment, we select 16 conventional demosaicing algorithms in Table 2-1. Note that these algorithms are highly diversified including the early non-adaptive algorithms as well as the state-of-art highly adaptive algorithms. We have also included two pairs of close demosaicing algorithms, i.e. Chang's and Wang's methods are extensions of Pei's and Kimmel's proposals, respectively. One can refer to [131-144] for details about these 16 methods used.

2.5.1 Weights Estimation

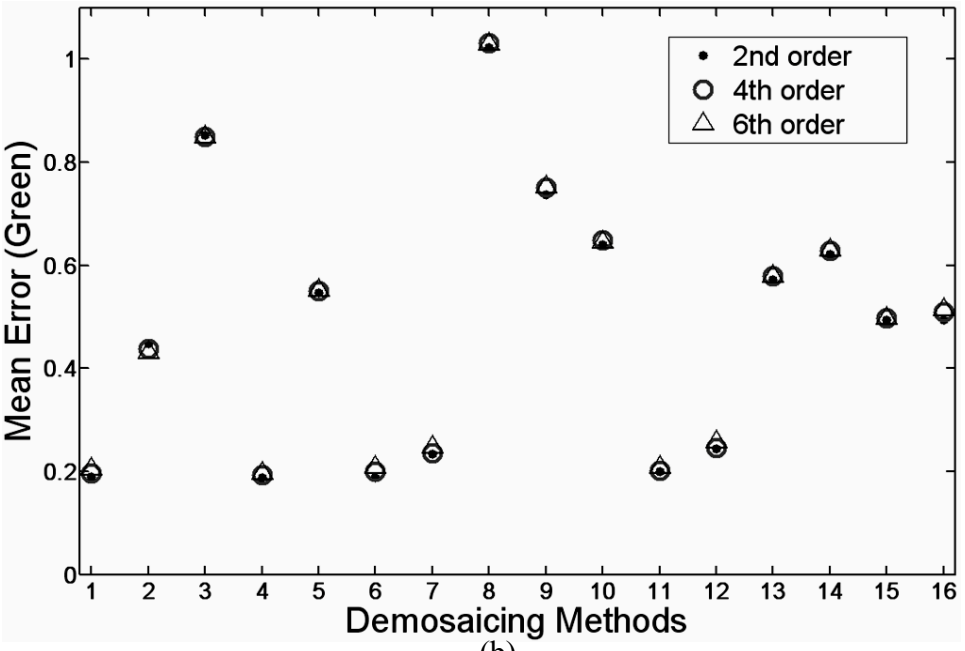
The proposed algorithm can be used to estimate the weight parameters associated with the applied demosaicing technique from the output image. Shown in Fig. 2-7 is an example, where estimated weights based on 4 different Bayer CFAs for the **G-on-*r*** *x*-axis demosaicing category are presented for a demosaiced color image using Hamilton's algorithm. From the results, we can see that the estimated weights in Fig. 2-7(c) match well with the ground-truth derivative weights in Fig. 2-7(b) in the estimation based on the correct Bayer CFA in Fig. 2-1(d). The small differences between the estimated weights and ground-truth weights are caused by the quantization distortion after the demosaicing process. We have also estimated the weights based on the 3 incorrect Bayer CFAs in Fig. 2-1 (e-g). The differences in the detection based on different CFAs lie in the different separation of sensor samples and the demosaiced samples. The corresponding results in Fig. 2-7(d-f) show that our estimated weights based on the incorrect CFAs are distinctively different from the estimated weights in Fig. 2-7(c) to suggest the detection statistics from the 4 channels of different CFAs are typically very different and forms unique patterns. This imbalanced pixel correlation patterns are good features for image forensics applications.

2.5.2 Re-estimation Accuracy

Our derived demosaicing weights together with Eqn (2.4) can be used to re-generate the demosaicing samples from the sensor samples. The re-estimation accuracy reveals the preciseness that our estimated weights match with the formulas used by the



(a)



(b)

Fig. 2-8 Comparison of Mean Absolute Re-Estimation Errors for Sixteen Demosaicing Methods Based on (a) Different Detection Algorithms and (b) the Proposed Algorithm with Different Approximation Orders. Refer to Table 2-1 for IDs of the Demosaicing Methods. The Mean Absolute Error is Computed When Each Color Sample is Represented by Eight Bits, i.e. from 0 to 255

underlying demosaicing algorithm. In this experiment, we compare our re-estimation accuracies for the sixteen demosaicing algorithms in Table 2-1 with the detection

algorithm [45] based on syntactic images. The syntactic set contains 1000 color images of 512×512, which are cropped from the center of 1000 photos taken with different cameras and then downscaled sufficiently to remove previous filtering characteristics. We first sample the RAWs from these images according to the first Bayer CFA in Fig. 2-1(d) and then separately demosaic these RAWs with the 16 demosaicing algorithms. Our proposed detection model is then applied to compute both the optimal weights and the corresponding errors for green channel of each demosaiced image. By comparing the difference of the absolute horizontal and vertical second-order gradients with a threshold, the work in [45] proposed to classify the image area into three regions, the horizontal-gradient, the vertical-gradient and the smooth regions. As mentioned early, this heuristic classification is rough and the results largely depend on the selected threshold. Through empirically determining a good threshold for this method and based on the outcomes of this rough reverse classification, our results in Fig. 2-8(a) show that the derivative-based correlation model in Eqn (2.6) is more accurate for most of the adaptive demosaicing methods that utilize cross-channel information to demosaic the green channel. If the proposed EMRC is applied, the mean absolute errors drop to a very low level for all the sixteen demosaicing algorithms. The comparison results suggest that our method achieves large margins of improvement on accuracies especially for the adaptive demosaicing methods, where the reconstruction is in cross-channel domains. Fig. 2-8(b) suggests that the 2^{nd}-, the 4^{th}- and the 6^{th}-order approximations give similar mean errors. Here, the different approximation orders refer to different ways of deriving our second-order derivative formula based on Taylor series expansion. In Fig. 2-9, we can see the iterations in the proposed EMRC algorithm monotonically reduce the average re-estimation error to a low level for two very different demosaicing methods. This demonstrates our proposed EMRC algorithm efficiently improves the detection accuracy for diversified demosaicing algorithms.

2.5.3 Classification of Demosaicing Algorithms

In this experiment, we classify the sixteen demosaicing algorithms using our proposed demosaicing features. Since demosaicing is not the last process in camera processing pipeline, our experiment also takes into account of the common camera post-demosaicing processes including quantization (QT), white balancing (WB), edge

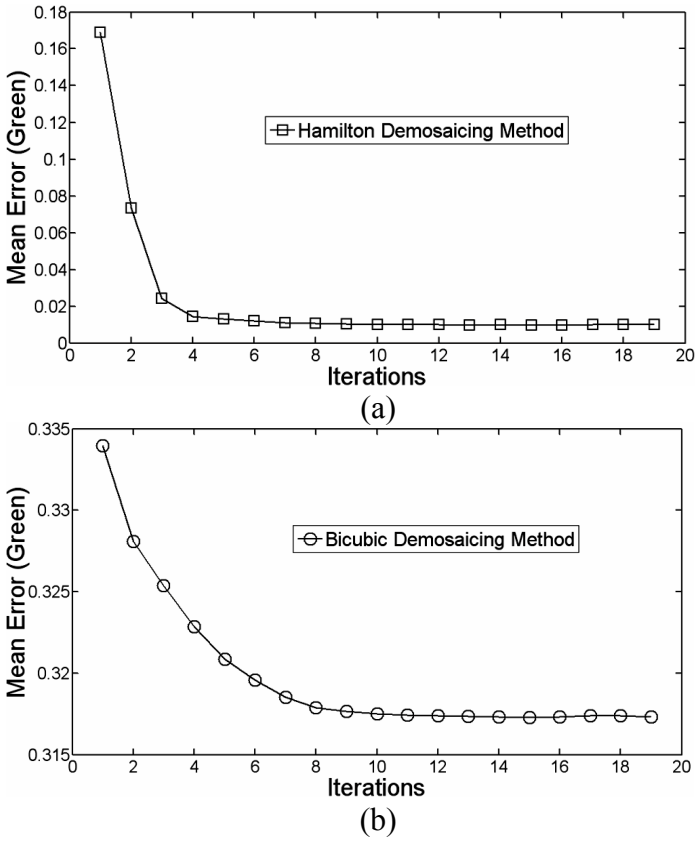
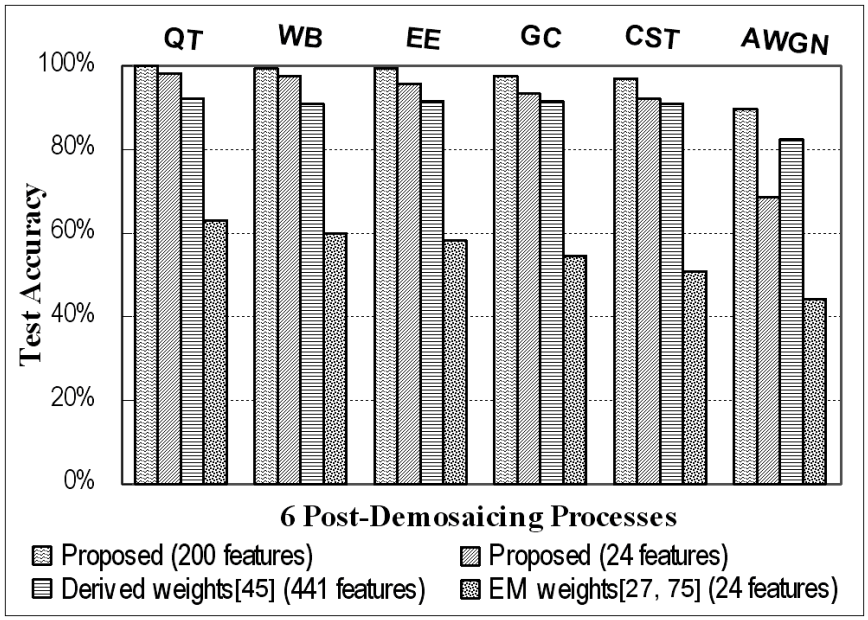


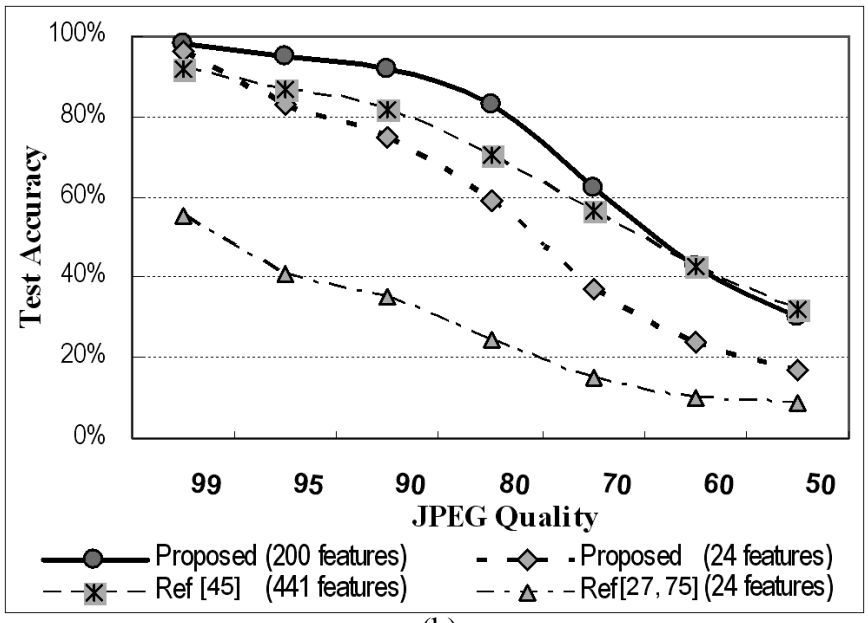
Fig. 2-9 Mean Absolute Prediction Error versus Iterations for (a) an Adaptive (Hamilton) and (b) a Non-Adaptive (Bicubic) Demosaicing Methods; (c) Test Image

enhancement (EE), gamma correction (GC), color space transformation (CST) and lossy JPEG compression. We also include adding zero-mean additive white Gaussian noise (AWGN) as one general form of distortion. The syntactic image set for this experiment is the same as the previous experiment. After demosaicing the 1000 images with these 16 algorithms, the demosaiced images further go through the above post-processing separately. Out of the 1000 syntactic images, 600 images per demosaicing algorithm and per distortion process are randomly selected for training and the rest for testing.

After performing feature extraction, we follow the LIBSVM guild [146] to train two probabilistic support vector machine (PSVM) classifiers with radial basis function (RBF) kernel. Before the training, we first linearly normalize each feature to the range of [-1, 1]. As the PSVM performance can be largely affected by the (C, g) parameter combination, we use the log-scale grid-search tool in LIBSVM toolbox to find the best



(a)



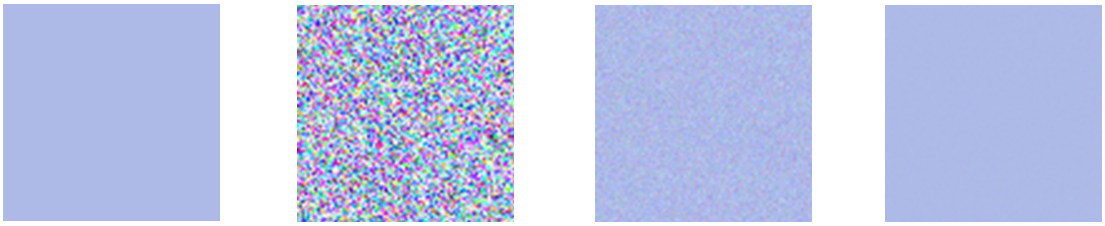
(b)

Fig. 2-10 Comparison of Source Demosaicing Algorithm Identification Using Various Demosaicing Features with Presence of (a) Six Common Post-Demosaicing Processes and (b) Lossy JPEG Compression of Various Quality Factors. In (a), the Post-Processes are: **QT**: Quantizing Color Samples of the Demosaiced Image to {0,1,…,255}; **WB**: White Balancing with 3×3 Transformation Matrix **T** = [.85 .07 .08; .01 .88 .11; .03 .08 .89]; **EE**: Edge Enhancement using Unsharp Mark (PSNR=30dB); **GC**: Gamma Correction with $1/\Upsilon=1/2.2$; **CST**: Color Space Transformation from CIEXYZ to sRGB; **AWGN**: Zero-Mean White Gaussian Random Noise (PSNR=30dB).

parameter combination before training the PSVM classifier. Since this grid-search on our large full training dataset can be very time consuming, we perform the grid-search

based on reduced training subset containing about 100 training images per class. Empirically, we find that our best parameter combination on the reduced training set has good generalization performance on the cross-validated training dataset. In the remainder of this thesis, the above processing procedures suggested in [146] are always performed as long as we use a PSVM classifier unless we state otherwise. For the six post-processes excluding JPEG, we use 57600 training and 38400 testing images. For JPEG of seven different quality factors, we use 67200 training and 44800 testing images. Based on the same setup, the comparison results in Fig. 2-10(a) demonstrate that our 200 demosaicing features selected by SFFS result in an average of 2.9% error rate for post-demosaicing processes including QT, WB, EE, GC, CST and AWGN, which reduce the error rates of 10.1% for the derived weights in [45] (441 features) by 2.5 times. With only 24 features selected, our average error rate increases to 9.1%, which is still 3.9 times smaller than 44.8%, the average error rate for the EM weights [27, 75] (24 features). We attribute the significant error rate reduction to our accurate detection model, which more comprehensively captures the differences between the close demosaicing methods and our features contain less content-dependant variations. Our good performance also suggests the uniqueness of demosaicing regularities is still largely preserved after the six common post-demosaicing distortion processes.

For lossy JPEG compression, the test accuracies in Fig. 2-10(b) monotonically decrease as the JPEG quality decreases. As expected, a low JPEG quality factor implies large quantization step sizes especially for the high-frequency DCT components. While demosaicing regularity contains rich high-frequency correspondence among neighboring color samples, low-quality JPEG compression can remove the uniqueness of the detected demosaicing regularity. Though JPEG compression could significantly deteriorate the classification performance, at a typical JPEG quality of 80, our 83% top-1 test accuracy based on 200 SFFS-selected features still suggests a reliable identification performance especially when multiple pairs of close demosaicing algorithms are present. The comparison results show that our average error rate of 28% based on 200 selected features is 17.6% lower than the average error rate for the derived weights in [45]. With the same number of features, our test accuracy based on 24 features confidently outperforms that of the 24 EM weights in [27, 75].



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

Fig. 2-11 (a) A Uniform 'Sky' Scenery (256×256) and After Adding AWGN Camera Sensor Noises of (b) PSNR = 10 dB; (c) PSNR = 30 dB and (d) PSNR = 50 dB

2.5.4 Classification of Post-Processes

Different types of post-demosaicing processes distort the demosaicing regularity in unique manners. In this experiment, with the demosaicing algorithm fixed as Hamilton's method [135], we classify seven post-demosaicing processes including quantization (eight-bit), WB, EE (PSNR=30dB using unsharp mark), GC ($1/\gamma=1/2.2$), CST (from CIEXYZ to sRGB), AWGN (30 dB) and lossy JPEG compression (Q-factor = 80) using the proposed features. With 600 randomly selected training images and 400 test images per distortion process, i.e. 4200 training and 2800 testing images in total for the seven distortions, we achieve a test classification accuracy of 97.3% using a similar PSVM classifier based on only 5 SFFS-selected features. This high classification rate shows that our proposed features also effectively capture the unique distortion characteristics of the different post-demosaicing processes. Since the post-processing in commercial DSCs can differ significantly in the type of inclusive processes, their sequence and the parameters, our good performance implies that the proposed features can also be used as a fingerprint of the post-demosaicing processing.

2.5.5 Sensitivity to Image Variations

Rich content variations in RAW samples make detection of the demosaicing regularity easier. In this simulation, we consider the extreme case of uniform scenery and the only image variations are contributed by sensor imperfections aggregated from sensor noises, dust and some processing distortions. We assume these noises are AWGN and the required noise level for reliable identification of the sixteen demosaicing algorithms based on our proposed features is studied.

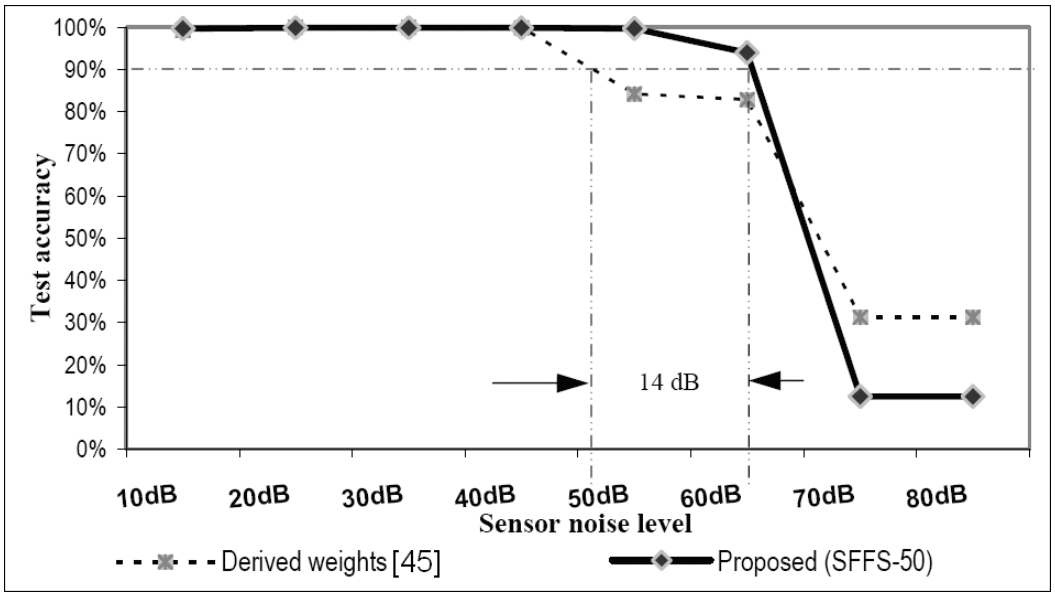


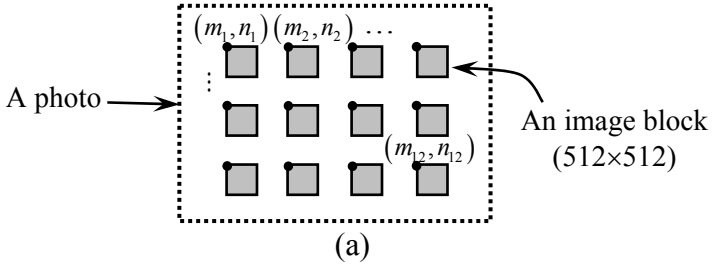
Fig. 2-12 Comparison of Test Accuracies in Classification of Sixteen Demosaicing Algorithms Based on Uniform Blocks with Various Sensor Noise Levels

We first create a noisy image set containing 1200 images by adding eight levels (PSNR=10, 20,…, 80dB) of random AWGN noises to the uniform "sky" scenery in Fig. 2-11(a). These noisy images are then separately demosaiced with the 16 algorithms. With a similar PSVM classifier and by selecting only 50 proposed features using SFFS, our comparison result in Fig. 2-12 shows that for a cut-off level of 90%, our required PSNR is 66dB, which is about 14 dB higher than [45]. This suggests that our proposed features require significant less scenery variations to achieve good source identification of demosaicing algorithms.

2.6 Forensics Source Identification

2.6.1 Camera Model Identification

The real camera processing pipelines in commercial DSCs contain proprietary knowledge. In this section, we further test the proposed demosaicing features in distinguishing 14 commercial DSC cameras in Table 2-2. We first establish a photo dataset consisting of 200 photos per camera. All these photos are the direct outputs from their cameras and are stored in the default JPEG formats. As illustrated in Fig. 2.13(a),



(a)

(b)

Fig. 2-13 Image Set Creation for Camera Identification; (a) Schematic Diagram of Cropping Blocks from a Photo, where Black Dots Indicate the Top-Left Corners of the Blocks, Which are Set to be Odd Number to Avoid Shifts to Underlying CFA; (b) Samples of Cropped Image Blocks

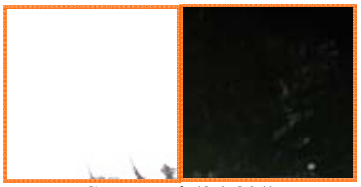
Table 2-2 Camera Models Used with 200 JPEG Photos for Each Camera

ID	Brand	Model	Photo Size
1	Canon	Ixus i	2272×1704
2		Powershot A620	3072×2304
3		EOS 400D	3888×2592
4		EOS 10D	3072×2048
5	Nikon	Coolpix S210	3264×2448
6		D70	3008×2000
7	Lumix	DMC-FX01	2816×2112
8		DMC-FX02	2304×1728
9	Olympus	U300d	2048×1536
10		E-500	3264×2448
11	Sony	DSC-P73	2304×1728
12		Alpha A350	4592×3056
13	Casio	EX-Z60	1600×1200
14	Fujifilm	FinePix Z2	2736×1824

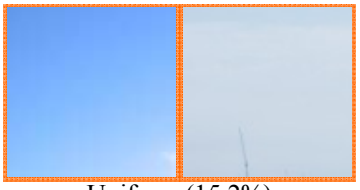
we crop 12 non-overlapped image blocks from each selected photo at 12 fixed locations to create 2400 image blocks per camera with some samples shown in Fig. 2-13(b). For each camera, we further divide the image blocks into a training set of 1584 blocks

Table 2-3 Confusion Matrix (%) for Fourteen-Camera-Model Classification (250 Features), Where Empty Fields Indicate Zeros

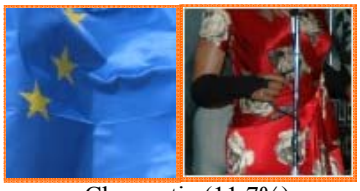
Ave. rate = 97.5%		Predicted camera													
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
Input camera	*1*	96.1	1.3	0.1	0.1		2.0		0.4						
	2	1.1	90.1	1.3			6.5	0.1	0.5						
	3	0.2		98.5	0.5		0.1	0.1				0.4	0.2		
	4	**Canon**			100	**Nikon**									
	5	0.1		0.4	0.5	98.0	0.1			0.2	0.1				0.6
	6	1.6	2.3		0.6	1.6	93.3	0.1	**Lumix**	0.3	0.1				0.1
	7			0.4			0.1	97.8		1.2	0.2		0.3		
	8	0.3	0.1	0.8	0.3		0.6		97.8						0.1
	9						0.6	0.1		99.3					
	10							0.1	**Olympus**	0.1	99.4		0.4		
	11			0.1	0.1		0.6	0.3			**Sony**	98.8	0.1		
	12					0.1	0.1	0.2	0.3		0.5		98.8		
	13						0.2			0.3			**Casio**	99.5	
	14	0.1	0.1			0.5	1.1				0.1			**Fujifilm**	98.0



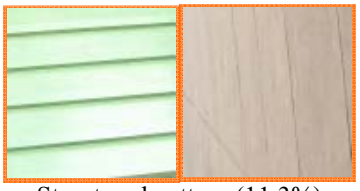
Saturated (34.3%)
Too dark or too bright



Uniform (15.2%)
Lack of content variation



Chromatic (11.7%)
Dominated by R, G or B color



Structured pattern (11.3%)
Lack of edge-type variations

Others (27.5%)

Fig. 2-14 Samples of Wrongly Classified Camera Image Blocks

cropped from 132 randomly selected photos and a test set of the remaining image blocks. For the 14 camera models, the number of blocks used for training and testing are respectively 22176 and 11424.

With a similar PSVM classifier, our identification results based on 250 selected features by SFFS in Table 2-3 demonstrate an excellent performance with an average test accuracy of 97.5%. Since the result is achieved based on individual blocks, the accuracy can be further improved to close to 100% simply by averaging the probabilistic scores of all 12 image blocks cropped from a test photo. The cameras we have tested

Table 2-4 RAW Tools Used with 200 TIFF Photos for Each RAW Tool

ID	RAW Tool	Image Size
1	ACDSee v10	3264×2448
2	Breeze Browser Pro v1.7	3337×2502
3	Capture One v3.7.8	3264×2448
4	Olympus Master v2	3264×2448
5	Photoshop CS2	3264×2448
6	Picture Window Pro v4.0	3340×2504
7	Rawshooter Essential 2006	3333×2499
8	Silkypix Developer Studio v3.0	3264×2448
9	StepOK v1.1	3340×2504
10	Corel Paintshop Pro 12	3336×2500

include multiple models from a same camera manufacturer, e.g. foru Canon models and two models each from Nikon, Lumix, Olympus and Sony. Our results in Table 2-3 shows the camera models from different manufacturers can be distinguished almost as accurately as the camera models from the same manufacturers.

With a close examination on the wrongly classified camera blocks, we found that the 2.5% total classification errors are mainly contributed by some low-quality blocks and we manually classify them into 5 categories as illustrated in Fig. 2-14. About 49.5% of the total errors are contributed by the "saturated" and the "uniform" categories. Images in these two categories lack content variations, which makes reliable detection of the demosaicing regularity almost impossible. The "chromatic" and the "structured" categories also contribute a small portion of errors because these images either lack color variations or edge-type variations which makes comprehensive detection of demosaicing regularity difficult. Though majority of the low-quality test blocks are still correctly classified, our finding suggests that the error rate can be further reduced if we wisely crop the image blocks at locations where rich color and edge information are present.

We have also recorded the computational time for the major training and testing processes in this identification experiment on a P4-2.66GHz PC. The average time taken for our demosaicing extraction on a 512×512 image using MATLAB v7.4 is about 45 seconds. After feature scaling, the SFFS feature selection time taken for the first pass using MATLAB v7.4 is about 45,007 seconds based on one training partition containing

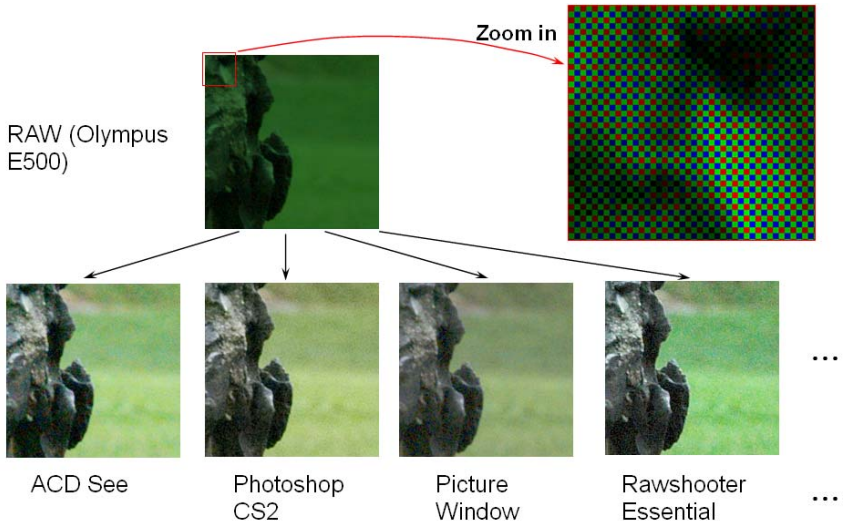


Fig. 2-15 Developing RAW into Photos Using Different RAW-Tools

Table 2-5 Confusion Matrix (%) for 10-RAW-Tool Classification (50 features), Where Empty Fields Indicates Zeros

Ave. rate = 99.1%		**Predicted RAW tool**									
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Input RAW tool	*1*	98.5	0.1	0.1		0.1		0.2		1.0	
	2		99.9						0.1		
	3			98.4	0.4		0.1		1.0	0.1	
	4				100						
	5	0.4		0.1		98.4		0.2	0.3	0.6	
	6	0.1					97.8		0.1	2.0	
	7				0.1	1.3		98.6			
	8	0.1			0.1	0.1	0.1		99.6		
	9						0.1			99.8	0.1
	10						0.1		0.1	0.2	99.6

1400 images, i.e. 100 images per DSC model. On average, each image requires about 32 seconds in the first-pass SFFS feature selection. Since we have used eight training partitions, the first-pass SFFS feature selection is repeated eight times before about 450 features are selected out of a total of 1536 demosaicing features. The second-pass SFFS takes 1,337,532 seconds to select 300 features based on one randomly selected training feature partition, indicating an average time of 955.4 seconds per image. For finding the best PSVM parameter combination, the grid-search time using the LIBSVM toolbox [147] based on 250 SFFS-selected features and one training partition is about 275 seconds or 0.20 second per image. With the best PSVM parameters (C=8, g=0.125), our PSVM training time based on the entire 22176 training images is about 214 seconds or

0.01 second per training image. The total testing time for 11424 testing images is about 61 seconds or 0.0053 second per testing image.

2.6.2 RAW-Tool Identification

Digital single-lens reflex (DSLR) cameras usually allow the sensor samples, the so-called "digital negatives" to be saved into RAW formats. As shown in Fig. 2-15, these RAWs can be developed into photos using various commercial RAW tools at a later time. These RAW tools are popularly used among the photo enthusiasts as they provide controllable environments to allow users experimenting on various darkroom parameters to achieve the most desirable effects. Similar to the camera processing, these tools demosaic the RAWs and then apply some post-demosaicing processes. However, since these RAW tools are designed for personal computers (PCs), common constraints such as processor speed, memory size and shot-to-shot delay are no longer among the key design considerations. Hence, these RAW tools are affordable to implement more complex demosaicing and post-processing to achieve higher image quality. In this test, we classify 10 commercial RAW tools in Tablet 2-4 with our proposed features.

We first develop 200 RAWs (captured by an Olympus E-500 camera) into uncompressed TIFF photos using the 10 commercial RAW tools in Table 2-4. Based on a similar setup to the camera identification experiment, we use a total of 15840 cropped image blocks for training and 8160 blocks for testing. We identify the source RAW tool using only 50 selected features based on cropped image blocks. Our average identification accuracy of 99.1% in Table 2-5 shows that the proposed detection framework can efficiently capture the unique processing characteristics of these RAW tools.

2.6.3 Analysis on Features Selected

Considering that the selected features largely depend on both the datasets used and the SFFS configurations, we show in Table 2-6 the more reliable distribution of the selected features. Among the 3 different types of demosaicing features, we find that for both camera and RAW-tool identifications, a dominant percentage of about 80% selected features are the weights features followed by the error cumulants features and

Table 2-6 Contribution Percentages (%) To The SFFS Selected Features from Three Different Feature Types and from Sixteen Different Demosaicing Categories (Refer To Fig 2-5 For the Demosaicing Categories)

Identification applications			**Camera Models**	**RAW Tools**
Feat. type	Weights		83	76
	Error Cumulants		14	24
	Normalized Group Sizes		3	0
Demosaicing category	**G on r**	x	5	6
		y	7	2
		o_{xy}	9	6
	G on b	x	4	0
		y	6	4
		o_{xy}	5	2
	R on g	x	5	8
		y	5	4
	R on b	u	7	12
		v	9	10
		o_{uv}	6	4
	B on g	x	5	6
		y	5	12
	B on r	u	7	6
		v	7	8
		o_{uv}	8	10

the least are the normalized group sizes features. Though partially due to the different population sizes of different feature types, these results also suggest that our estimated weights contain the richest discriminant information. The contribution to the selected feature set from the sixteen demosaicing categories is rather even. We also note that 36% of selected features are contributed from the green demosaicing categories for camera model identification and only 20% for the RAW-tool identification. Considering the major difference is that all camera photos in our experiment are in JPEG format and the converted RAW photos are in TIFF format, the difference in contribution percentages from the green categories is likely caused by JPEG compression. In the TIFF photos, we observe in many cases that our detected demosaicing regularity from the different green categories tend to be more similar, exhibiting less diversity information. Hence, fewer features are selected from the green categories.

2.7 Summary

In this chapter, we present an accurate detection framework of image demosaicing regularity. Our proposed algorithm addresses the common differences of color filtering and demosaicing algorithms, and the accurate detection is achieved by a precise reverse classification together with partial derivative correlation models. In the reverse classification, an EMRC algorithm is demonstrated to be effective in resolving the ambiguous demosaiced axes. Through using partial derivative correlation models, our method efficiently detects both the cross-channel and the intra-channel correlation caused by demosaicing. By significantly reducing our feature dimension using sequential forward floating search, the simulation results show that our proposed demosaicing features confidently outperform two existing demosaicing detection methods in identifying sixteen demosaicing algorithms in the presence of various common post-demosaicing processes. Our proposed features are also highly effective in distinguishing different post-processes and are more sensitive to small scenery variations. When applied to real forensics applications, our reduced sets of demosaicing features achieve nearly perfect average test identification accuracies of 97.5% for fourteen commercial DSCs of different models and 99.1% for ten RAW-tools based on large number of cropped image blocks, which contain a mixture of good and bad quality images.

Chapter 3 Mobile Camera Model Identification through Eigenfeature Regularization and Extraction

Mobile cameras are typically low-end cameras equipped on handheld devices such as personal digital assistants and cellular phones. The fast proliferation of these mobile cameras has brought up concerns on the origin and integrity of their output images. On another hand, high feature dimensionality and limited training data has been a common problem in using large number of diversified forensics features for reliable forensics analysis. In this chapter, we identify the source mobile cameras by joining three types of demosaicing features extracted from the output mobile images. Through eigenfeature regularization and extraction, comparison results show that our joined feature set performs significantly better than a number of conventional types of statistical image features in distinguishing nine mobile cameras of dissimilar models. In a fifteen-cam classification, our joined feature set achieves excellent test accuracies in distinguishing cameras from different brands and dissimilar models and the test accuracies tend to mix among cameras of the same or very similar models.

3.1 Introduction

Since the first commercial camera phone (J-SH04 by Sharp) was made in 2000, mobile cameras incorporated in the handheld devices have experienced tremendous growth. Nowadays, over 90% of available handsets have mobile cameras attached [8].

Though the quality of these mobile camera are hardly comparable with that of the digital still cameras (DSC), the resolution and image quality of these mobile cameras have been steadily improving so that the technological gap between mobile cameras and DSCs is constantly shortening. The application prospect of these mobile cameras is highly promising and their mobile photos are popularly shared on Internet (e.g. photo blogs) and the newsworthy pictures can even be accepted and published in news reports to make great impact. This has also brought up serious concerns on the origin and integrity of these low-end mobile pictures.

In recent years, passive image forensics has been extensively studied mainly for high-quality photos from DSCs. Though mobile cameras in general share a similar processing pipeline in Fig. 1-5 as the DSCs, it is worth to note that, a mobile camera is usually about ten times cheaper, ten times smaller in size (both sensor chip and camera head) and consume ten times less power than a DSC [8]. Under such constraints, capturing pictures of descent quality requires special software processing including strong denoising, crude demosaicing and white balancing algorithms and JPEG compression. As some special processes may render certain unique high-frequency regularities undetectable, not all passive image forensics methods for DSCs can be readily extended to mobile cameras. Several existing forensics works for mobile photos includes: By extending the identification technique developed for DSCs in [34], Alles *et al.* [52] detected the PRNU sensor noise pattern to identify the individual mobile cameras and webcams from their output images. McKay *et al.* [57] computed both color interpolation coefficients (CIC) and some noise statistics (NStats) features to identify the image acquisition devices including cellular cameras and scanners. Tsai *et al.* [46] combined several sets of statistical image features including color features, image quality metrics (IQM) and wavelet features to identify both DSCs and mobile cameras. Similar to [46], Celiktutan *et al.* [49] used various fusion methods to combine three sets of statistical features including IQM, wavelet statistics (WS) and binary similarity (BS) features to distinguish cell-phone models based on typically low-resolution mobile images. With 192 features selected using sequential forward feature selection, their work achieves 95.1% accuracy with feature-level fusion in identifying sixteen mobile cameras including one pair of mobile cameras of the same model.

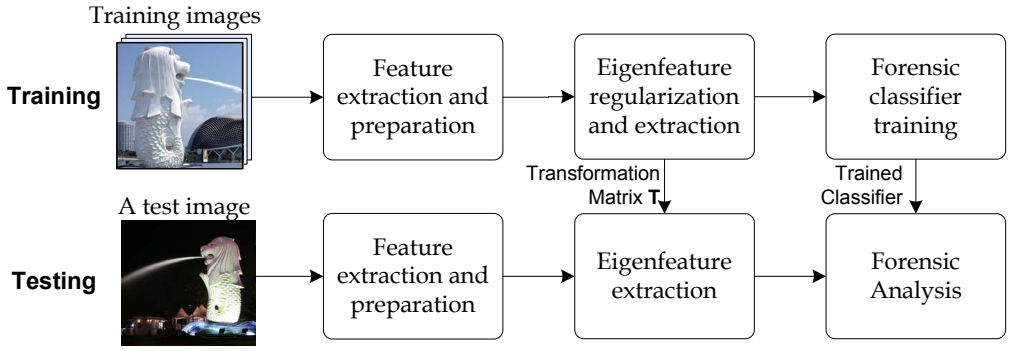


Fig. 3-1 Overview of Our Proposed Forensics Analysis Framework

Our work in Chapter 2 accurately estimates the underlying demosaicing formulas from demosaiced images through a precise reverse classification and partial second-order image derivative correlation models. Since mobile cameras are also single-sensor based and they rely on color filtering and demosaicing techniques to economically produce color, we propose to extend our demosaicing features computed in Chapter 2 to mobile image forensics. These features characterize both the applied demosaicing algorithm and the post-demosaicing processing. Through eigenfeature regularization and extraction, we demonstrates experimentally that our low-dimensional eigen demosaicing features show superior results in identification of the source mobile cameras based on the cropped image blocks as compared to the same number of eigen features extracted from other statistical forensics features.

The rest of this Chapter is organized as follows. Section 3.2 details the proposed framework where the eigenfeature extraction technique used is highlighted. Section 3.3 experimentally demonstrates the effectiveness of using the extracted eigenfeatures for mobile camera identification. Section 3.4 summarizes this chapter.

3.2 Proposed Forensics Framework

Shown in Fig. 3-1 is the overview of our proposed forensics framework. In the training phase, diversified forensics features are extracted from a given set of training images and these features are fused at feature level to form the overall feature vectors. Each overall feature vector could contain a same large number of features. We propose to apply an eigenfeature regularization and extraction technique to transform the overall

feature vectors into compact eigenfeature vectors, which are used to train a forensics analyzer. In the testing phase, the overall test feature vector extracted is transformed into a test eigenfeature vector using a transformation matrix **T** obtained in the training phase. At the end, the forensics conclusion is made by the trained forensics analyzer based on the test eigenfeature vector. In the following section, we elaborate the main steps in greater details.

3.2.1 Forensics Feature Extraction and Preparation

We extract a total of three diversified sets of demosaicing features. By following our demosaicing detection technique in Section 2.4, we obtain 1248 weights (WT) features, 256 error cumulants (EC) and 32 normalized group sizes (NGS) features. The different sets of features extracted from an image are fused at feature level to have an overall feature vector containing 1536 measurements. Based on the training feature set, we compute the global mean and the standard deviation of each individual feature so that we can linearly normalize each feature to have zero mean and unity variance.

3.2.2 Eigenfeature Regularization and Extraction

High feature dimensionality and relatively small set of training data are common problems in pattern classification. It is often desirable to extract a compact set of highly discriminant features through subspace transformation before the features are used in forensics analysis. In this work, we apply an eigenfeature regularization and extraction technique (ERE) [114] to generate a compact set of forensics features. This ERE method can reliably identify a discriminative subspace from an entire high-dimensional eigen space. Extensive experiments in [114] have demonstrated that the ERE method has better classification performance to the unseen data than many existing subspace methods.

Let $\{\mathbf{x}_{ij}\}$ denote the normalized training feature vectors, where $i \in \{1,\ldots,L\}$ and L is the total number of classes. $\mathbf{x}_{ij} \in \mathbb{R}^N$ and N=1536 is our feature dimension. Let the i^{th} class have Q_i number of training samples. The ERE method can be summarized below:

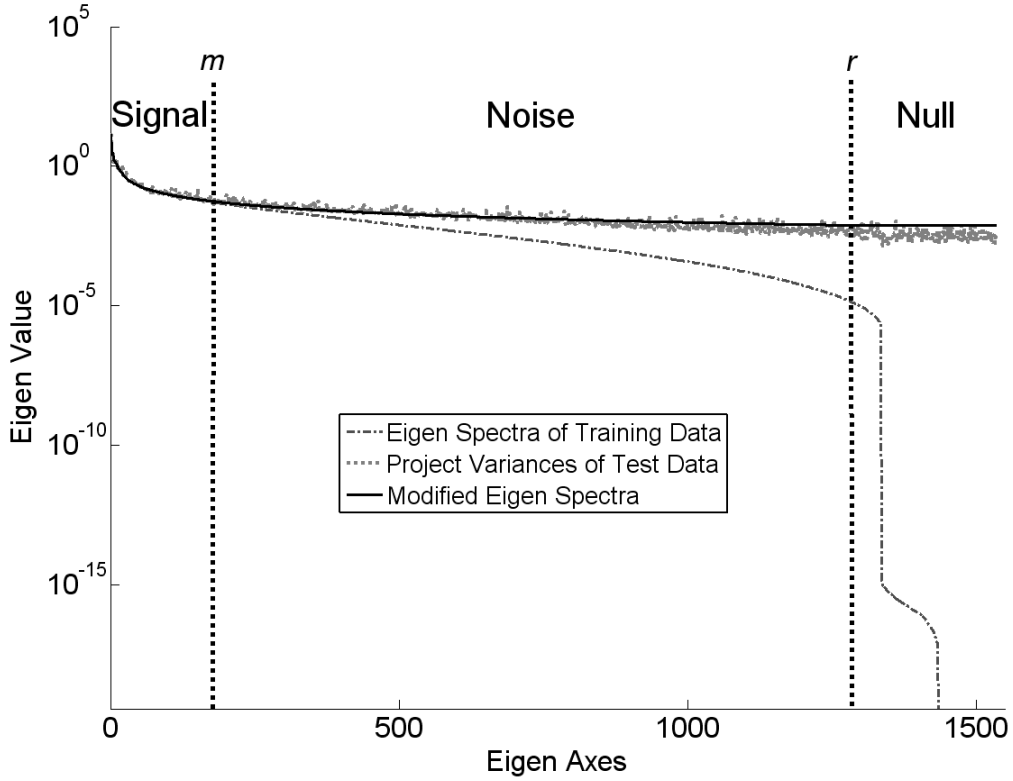


Fig. 3-2 Eigen Spectra of a Training Mobile Image Feature Set Containing Nine Classes of Mobile Cameras, Projected Variances of the Corresponding Test Mobile Feature Set and the Modified Eigen Spectra

1. Assuming each class has a equal prior probability of $1/L$, compute the within-class covariance matrix $\mathbf{S}^{(\mathrm{w})}$ using

$$\mathbf{S}^{(\mathrm{w})} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{Q_i}\sum_{j=1}^{Q_i}\left(\mathbf{x}_{ij}-\overline{\mathbf{x}}_i\right)\left(\mathbf{x}_{ij}-\overline{\mathbf{x}}_i\right)^T \tag{3.1}$$

2. Perform eigen decomposition using

$$\mathbf{\Lambda}^{(\mathrm{w})} = \mathbf{\Phi}^{(\mathrm{w})T}\mathbf{S}^{(\mathrm{w})}\mathbf{\Phi}^{(\mathrm{w})} \tag{3.2}$$

where $\mathbf{\Phi}^{(\mathrm{w})} = \left[\boldsymbol{\varphi}_1^{(\mathrm{w})},\ldots,\boldsymbol{\varphi}_N^{(\mathrm{w})}\right]$ is the eigenvector matrix of $\mathbf{S}^{(\mathrm{w})}$, and $\mathbf{\Lambda}^{(\mathrm{w})}$ is the diagonal matrix with eigen values $\lambda_1^{(\mathrm{w})},\ldots,\lambda_N^{(\mathrm{w})}$ corresponding to the eigenvectors. After sorting the eigenvectors according to descending order of the corresponding eigen values, we plot the eigen spectra shown in Fig. 3-2. It should be noted that the

enlarged gap in the log-scale plot can be observed between the eigen spectra and the test spectra. This suggests that the reliability of the eigen values estimated start degrading especially when the estimated eigen values becomes smaller and smaller. One way to increase the reliability of the small estimated eigen values is to significantly increase the amount of training data, e.g. to make the number of training images per class to be ten to twenty times of the feature dimension. However, practically this requirement can be hardly met especially when a large number of diversified forensics features are used.

3. Separate the eigen spectra into three regions, "**signal**", "**noise**" and "**null**" as illustrated in Fig. 3-2, and fit a regularized eigen spectra as [114]

$$\tilde{\lambda}_n^{(\mathrm{w})} = \begin{cases} \lambda_n^{(\mathrm{w})}, & n < m \\ \alpha/(n+\beta), & m \le n \le r \\ \alpha/(1+r+\beta), & r < n \le N \end{cases} \tag{3.3}$$

where n is a running index, $\lambda_n^{(\mathrm{w})}$ is the n^{th} largest eigen value of $\mathbf{S}^{(\mathrm{w})}$ and α, β are empirical constants. Parameters m and r are the indices defining the boundaries between the "signal" and "noise" regions and between the "noise" and "null" regions, respectively. The details of the choices of m, r, α and β can be found in [114]. From Fig. 3-2, we can see that the regularized eigen spectra is closer to the test variance curve in the "noise" region and the unreliable spectra in the "null" region are replaced by a constant spectra.

4. Perform the whitening feature transformation based on the regularized eigen spectra

$$\mathbf{y}_{ij} = \tilde{\boldsymbol{\Psi}}_N^{(\mathrm{w})T} \mathbf{x}_{ij} \tag{3.4}$$

where $\tilde{\boldsymbol{\Psi}}_N^{(\mathrm{w})} = \left[\boldsymbol{\varphi}_1^{(\mathrm{w})} \Big/ \sqrt{\tilde{\lambda}_1^{(\mathrm{w})}}, \ldots, \boldsymbol{\varphi}_N^{(\mathrm{w})} \Big/ \sqrt{\tilde{\lambda}_N^{(\mathrm{w})}} \right]$.

5. Based on the $\{\mathbf{y}_{ij}\}$, compute the total covariance matrix

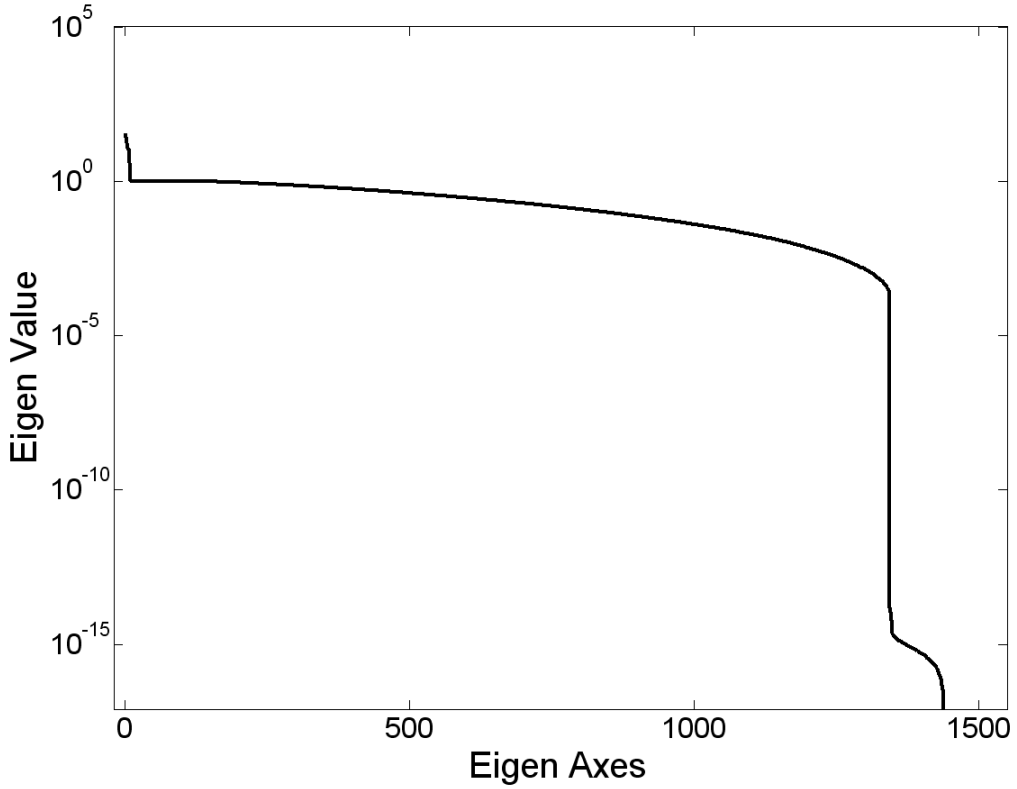


Fig. 3-3 Eigen Spectra of the Total Covariance Matrix after the Whitening Transformation

$$\mathbf{S}^{(t)} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{Q_i}\sum_{j=1}^{Q_i}\left(\mathbf{y}_{ij} - \overline{\mathbf{y}}\right)\left(\mathbf{y}_{ij} - \overline{\mathbf{y}}\right)^T \tag{3.5}$$

6. Perform eigen decomposition on $\mathbf{S}^{(t)}$ similarly to Eqn (3.2) and sort the eigen vectors $\tilde{\boldsymbol{\psi}}^{(t)} = \left[\boldsymbol{\varphi}_1^{(t)}, \ldots, \boldsymbol{\varphi}_N^{(t)}\right]$ corresponding to their eigen values in descending order. From the eigen spectra of $\mathbf{S}^{(t)}$ in Fig. 3-3, we can see that most of the eigen values in the middle range are close to 1 and this is due to the whitening transformation in Eqn (3.4). A few leading Eigen values are significantly larger than 1 to indicate that the dominant between-class variances are present on these eigen axes and these axes can be combined to form a good discriminant subspace for classification.
7. Construct $\tilde{\boldsymbol{\psi}}_D^{(t)} = \left[\boldsymbol{\varphi}_1^{(t)}, \ldots, \boldsymbol{\varphi}_D^{(t)}\right]$ for dimensionality reduction, where typically $D \ll N$, $\boldsymbol{\varphi}_1^{(t)}, \ldots, \boldsymbol{\varphi}_D^{(t)}$ are the D leading eigen vectors of $\mathbf{S}^{(t)}$ corresponding to the D largest eigen values. The compact feature vector

$$\mathbf{z}_{ij} = \tilde{\boldsymbol{\psi}}_D^{(\mathrm{t})T} \mathbf{y}_{ij} = \mathbf{T}\mathbf{x}_{ij} \tag{3.6}$$

where the overall transformation matrix $\mathbf{T} = \left(\tilde{\boldsymbol{\psi}}_N^{(\mathrm{w})} \tilde{\boldsymbol{\psi}}_D^{(\mathrm{t})}\right)^T$ and $\mathbf{z}_{ij} \in \mathbb{R}^D$. In the testing procedure in Fig. 3-1, the matrix $\mathbf{T}$ can be directly used to transform an overall test vector into a compact eigenfeature vector.

Through regularizing the "noise" and "null" regions of the within-class eigen spectra higher, the risks of using the inverse of these eigen values in the whitening transformation in Eqn (3.4) is being reduced and the reliability of this ERE method in finding a good discriminant subspace is increased. At the same time, since the good discriminant eigen axes reside in all the three eigen spectra regions, the ERE method also ensures that some good discriminant eigen axes in the "noise" and "null" regions are not prematurely removed.

3.2.3 Forensics Classification

With the compact set of eigenfeature vectors obtained from different forensics classes, a forensics classifier can be readily trained for making forensics analysis. In this work, we train both a simple first nearest neighbor classifier (1NNK) and a sophisticated probabilistic support vector machine (PSVM) classifier with radial basis function kernel to identify *L* source mobile cameras.

For the 1NNK classifier, a cosine distance measure between a probe feature vector $\mathbf{z}_p$ and a gallery feature vector $\mathbf{z}_g$ is given below

$$D_{\cos}\left(\mathbf{z}_{\mathrm{p}}, \mathbf{z}_{\mathrm{g}}\right) = -\mathbf{z}_{\mathrm{p}}{}^T \mathbf{z}_{\mathrm{g}} \Big/ \left(\left\|\mathbf{z}_{\mathrm{p}}\right\| \cdot \left\|\mathbf{z}_{\mathrm{p}}\right\|\right) \tag{3.7}$$

where $\|\cdot\|$ denotes Frobenius norm. The gallery feature vector of a forensics class is computed by simply averaging all the training feature vectors from the class.

For the PSVM classifier, we use the tools provided in LIBSVM [147] and follow the guild in [146] to obtain the desired classification results.

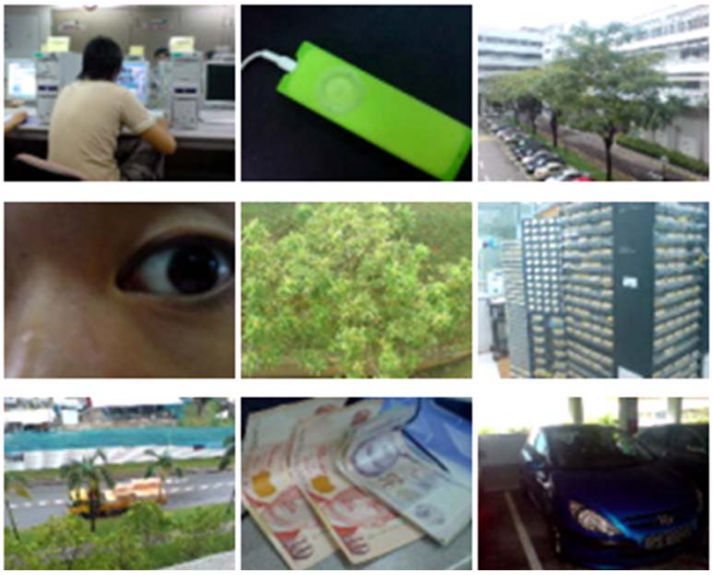
Table 3-1 Mobile Cameras Used with 100 Pictures from Each Camera

ID	Brand	Model	Native Resolution
N1	Nokia	3230	1280×960
N2		3250	1600×1200
N3		5300	1280×960
N4		6280	1600×1200
N5		7390	2048×1536
N6		7390	2048×1536
N7		N73	2048×1536
N8		N73	2048×1536
N9		N73	2048×1536
S1	Sony Ericsson	K750	1632×1224
S2		K750	1632×1224
S3		K800	2048×1536
S4		W800	1632×1224
L1	LG	KG320	960×1280
O1	O2	XDA Atom	1600×1200

3.3 Experimental Results and Discussion

To test the efficacy of our proposed features, we have set up a mobile photo set containing 1500 photos from a total of 15 mobile cameras in Table 3-1, where cameras of identical or very close models are present. These photos are collected from 15 contributors by their mobile cameras and all photos are direct camera output stored in the default JPEG format. The default photo sizes of different camera models vary from 1280×960 to 2048×1536. These photos cover a large variety of common indoor and outdoor scenes captured under different lighting conditions. Several representative picture samples from two cell-phones are shown in Fig. 3-4. We crop four non-overlapping blocks of about 512×512 at fixed locations from each photo to get 400 blocks per camera. For each camera, we randomly apportion the images into a training set of 300 blocks cropped from 75 mobile photos and a test set of the remaining blocks. The random apportion is repeated for another four times so that we have five different combinations of the training and test image sets.

(a)

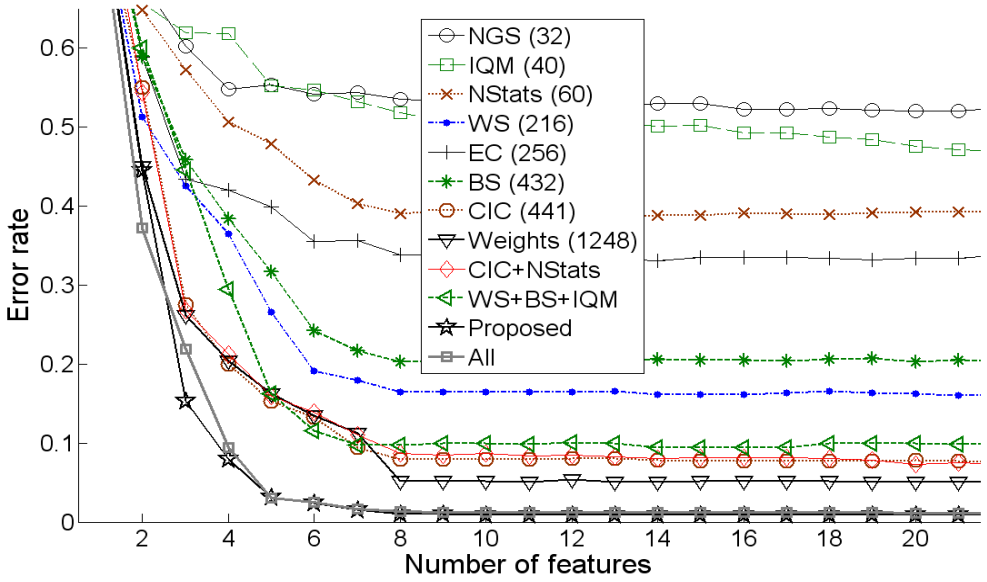


(b)

Fig. 3-4 Sample Mobile Images from (a) a Nokia 7390 Cell-Phone and (b) a Sony Ericsson K800 Cell-Phone

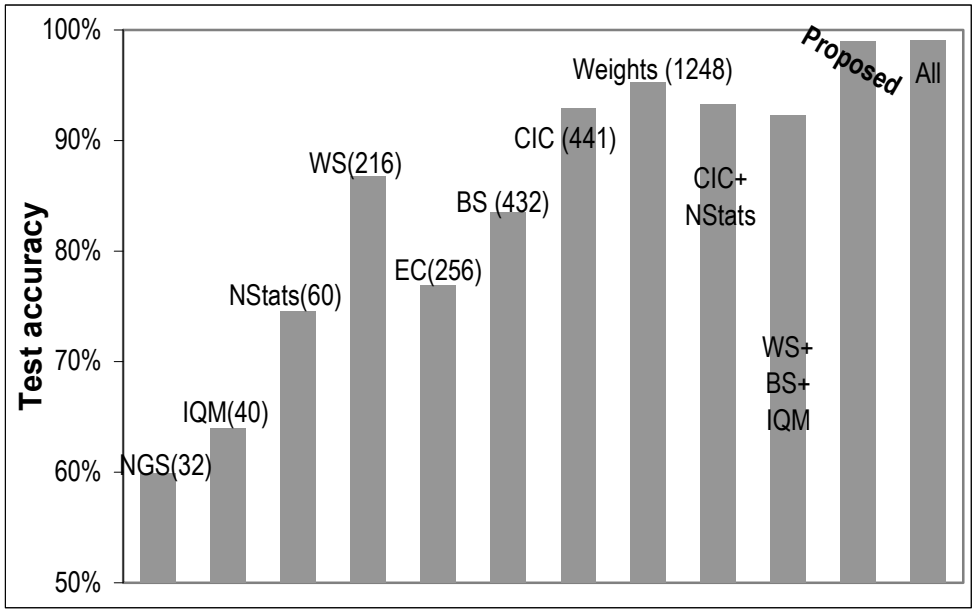
3.3.1 Comparison for Nine-Camera-Model Classification

Our proposed features are associated with camera software processing, which are known to be similar for mobile cameras of the same model. In this experiment, we test our proposed features for identification of nine cameras of dissimilar model labels including N1, N3, N4, N5, N7, S1, S3, L1 and O1 in Table 3-1. By applying the same



NGS: Normalized Group Sizes
IQM: Image Quality Metrics
NStats: Noise Statistics
WS: Wavelet Statistics
EC: Error Cumulants
BS: Binary Similarities
CIC: Color Interpolation Coefficients
Proposed: NGS+EC+Weights
All: All Different Features Combined

(a)



(b)

Fig. 3-5 Comparison of Various Feature Sets in Nine-Cam-Model Identification with the Number Inside Parentheses Indicating Dimension of the Original Feature Set; (a) Average Error Rates Vs Number of Eigen Features (1NNK Classifier with Cosine Dissimilarity Measure are Used); (b) Average Test Accuracy Achieved using PSVM Classifier and RBF Kernel with 20 Eigen Features

ERE technique and based on the same number of eigen features, we compare in Fig. 3-5 our demosaicing features with several state-of-the-art statistical image forensics features

in literature for mobile camera identification. These feature sets include the wavelet statistics (WS) features [49, 62, 64], binary similarity measures (BS) [49, 100], image quality metrics (IQM) [49, 68, 77, 99], color interpolation coefficients (CIC) [45, 57], noise statistics (NStats) [57], the combination of WS, BS and IQM [49] and the combination of CIC and NStats [57]. For the BS features, we compute a total of 432 features based on the description in [49]. Though this number is still less than the 480 BS features used in [49], our BS feature set still covers majority of the BS features. With a simple 1NNK classifier and based on cosine dissimilarity measure, the comparison result in Fig. 3-5(a) shows that our proposed combination of Weights, EC and NGS works well and for all feature sets, the test error rate drops sharply initially and stabilizes to a low level after selecting about eight eigen demosaicing features.

With 20 eigen features and a more sophisticated probabilistic support vector machine (PSVM) classifier [147], the source identification accuracy is compared in Fig. 3-5(b). From this figure, our Weights features are the best performing type of features, whose test accuracy of 95.2% is 2.3% higher than that of the CIC features, 8.5% higher than WS and 11.8% higher than BS. Our proposed combination of demosaicing features achieve an average test accuracy of 99.0%, which is 6.7% higher than the combination of WS, BS and IQM features and 5.7% higher than the combination of CIC and NStats features. It should be noted that the test accuracy achieved by our joined demosaicing features is significantly better than that of the weights features alone. This shows that our EC and NGS features have good complementary effects to the weight features. If we combine all individual feature sets including WS, BS, IQM, NGS, EC, Weights, CIC and NStats and apply the same ERE feature reduction to 20 eigen features, the obtained test accuracy (the last bar in Fig. 3-5 (b)) is only 0.05% better than that of the proposed combination of demosaicing features. This suggests that our good results can hardly be further complemented by adding the several conventional feature sets. With an assumption that the ERE process and the PSVM classifier have no bias, our proposed demosaicing features outperform the conventional forensics features in the context of identifying mobile cameras of different models.

Ave. Model Accuracy = 94.8		N1	N2	N3	N4	N5	N6	N7	N8	N9	S1	S2	S3	S4	L1	O1
Nokia	3230 (N1)	100														
	3250 (N2)		98.4						0.4			0.8		0.4		
	5300 (N3)			100												
	6280 (N4)		0.4		98.4	0.8	0.4									
	7390 (N5)					90.8	2.0	1.6	3.6	2.0						
	7390 (N6)					1.6	98.4									
	N73 (N7)					2.8		74.0	8.8	14.4						
	N73 (N8)	0.4				1.2		12.8	80.8	4.8						
	N73 (N9)					3.6		13.6	4.8	78.0						
Sony Ericsson	K750 (S1)										80.0	13.6		6.4		
	K750 (S2)										5.2	90.4		4.4		
	K800 (S3)					0.4	2.8	0.8	0.4			0.8	92.8			
	W800 (S4)										16.4	23.2	0.4	60.0		
LG	KG320 (L1)						1.2								98.8	
O2	XDA Atom (O1)						0.4									99.6

Fig. 3-6 Confusion Matrix (%) of Source Identification for Fifteen Mobile Cameras, where the Empty Fields Indicate Zeros, Bold-Face Single Brackets Highlight Cameras of the Same Models and the Shaded Fields are the Identification Results among Cameras of the Same Models

3.3.2 Fifteen-Camera Identification

In this experiment, we derive 20 eigen features from our combined demosaicing features and use PSVM classifier to identify all 15 mobile cameras listed in Table 3-1. The average results from five different apportions are presented in Fig. 3-6 in terms of a confusion matrix, where its $(i, j)^{th}$ element is the probability of identifying the images from the i^{th} input camera as the j^{th} camera. The result demonstrates that our demosaicing features are highly efficient in distinguishing different mobile camera brands as well as dissimilar models of the same brand. The average test accuracies achieved in identification of the 4 brands and the 11 models are respectively 99.4% and 94.8%, which are highly promising. For cameras of the same model or very close models, since the software processing is identical or very similar, our test accuracies expectedly tend to mix with each other to certain extent depending on the camera models.

3.3.3 Eleven Camera-Model Identification

As discussed earlier, our demosaicing features are appropriate for distinguishing the software processing pipelines with different demosaicing processes or post-demosaicing

Ave. Model Accuracy=94.5		N1	N2	N3	N4	N5-6	N7-9	S1-2	S3	S4	L1	O1
Nokia	3230 (N1)	100										
	3250 (N2)		98.4	0.4			1.2					
	5300 (N3)			100								
	6280 (N4)				98	1.6	0.4					
	7390 (N5-6)					96.4	3.6					
	N73 (N7-9)					2.5	97.5					
Sony Ericsson	K750 (S1-2)							93.4		6.6		
	K800 (S3)					3.2	4.4	0.4	92			
	W800 (S4)							40.8		59.2		
LG	KG320 (L1)					1.2					98.8	
O2	XDA Atom (O1)						0.4					99.6

Fig. 3-7 Confusion Matrix (%) of Source Identification for Eleven Mobile Camera Models, where the Empty Fields Indicate Zeros, Bold-Face Single Brackets Highlight Cameras of the Same Models and the Shaded Fields are the Identification Results among Cameras of the Same Models

processing. Our previous experiment in Fig. 3-6 also shows that cameras of the same model are more likely to confuse with each other. In view of the large similarity of our demosaicing features for cameras within a same model, in this experiment, we group the mobile cameras of each model as one class and learn the eigenfeature transformation based on a total of eleven model classes. With the learned eigenfeature transformation, we subsequently derive 20 eigenfeatures and use them in conjunction a PSVM classifier to identify the eleven camera models. From the confusion matrix results in Fig. 3-7, we can see that most of the camera models can be identified with more than 96% accuracy except the camera models of Sony Ericsson brand. Especially, 40.8% test images of Sony Ericsson W800 model has been wrongly identified as another Sony Ericsson model, K750 and 6.6% of test images from K750 are wrongly identified as W800 model, which are the two largest error percentages. We conjecture that the two mobile camera models K750 and W800 implement an identical software processing pipeline. Though different mobile device models usually mean some differences in the mobile devices, these differences may lie in some non-camera related functional components for K750 and W800 models. Here, we shall note that a mobile device typically possesses many different functional components, where the attached camera is only one of the components. In the case that two mobile device models share an identical camera

Ave. rate = 98.0%		Predicted camera													
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
Input camera	*1*	97.9	0.7		0.1	0.1	1.0		0.1						0.1
	2	2.7	88.7				8.0		0.4	0.1					0.1
	3	Canon		99.5	0.3					0.1	0.1				
	4	0.1	0.1		99.7	Nikon							0.1		
	5	0.1		0.1	0.6	98.6	0.1				0.5				
	6	0.2	5.5		0.5	0.4	93.3	Lumix	0.1						
	7						0.1	99.6		0.1			0.1		
	8	0.3	0.1				0.9		98.2	0.1	0.1				0.3
	9	0.3	0.1				0.1			99.5					
	10								Olympus		99.9	0.1			
	11	0.4	0.1	0.3	0.1	0.2	0.1					98.7		0.1	
	12		0.1								Sony		99.9		
	13	0.1							0.1		0.1		Casio	99.7	
	14	0.3	0.1			0.1	0.1							Fujifilm	99.4

(a)

Ave. rate = 99.7%		Predicted RAW tool									
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Input RAW tool	*1*	99.7	0.3								
	2		99.9						0.1		
	3			99.8		0.1			0.1		
	4			0.1	99.6				0.3		
	5		0.5			99.4			0.1		
	6	0.1					99.3			0.6	
	7			0.1		0.1		99.8			
	8			0.1					99.9		
	9	0.1	0.1				0.4			99.4	
	10					0.1			0.1		99.8

(b)

Fig. 3-8 Confusion Matrices (%) of Source Identification for Fourteen DSC Models in (a) and for Ten RAW Tools in (b), where the Empty Fields Indicate Zeros, Bold-Face Single Brackets Highlight Cameras of the Same Models and the Shaded Fields are the Identification Results among Cameras of the Same Models. Refer to Table 2-2 for the fourteen DSC models and refer to Table 2-4 for the ten RAW tools

processing pipeline, our demosaicing features are expectedly to be ineffective to distinguish such models. This would be analogous to using our demosaicing features for identifying the individual camera devices within a same model.

3.3.4 DSC Model and RAW Tool Identification

Though we have used different feature reduction techniques, i.e. SFFS feature selection in Chapter 2 and ERE in Chapter 3, we do not imply that ERE is more appropriate for mobile camera model identification or SFFS is more suitable for identification of DSCs and RAW tools. In this experiment, we also test using ERE feature reduction technique in identification of the fourteen DSCs and the ten RAW tools based on the same image datasets we used in Chapter 2. By first learning the eigenfeature transformation for each task, we extract 20 eigen demosaicing features per image to train two PSVM classifiers, one for the DSC model identification and one for the RAW-tool identification. Based on similar plots to Fig. 3-5(a), we find that at least about 13 and 7 features are required for the identification tasks on DSCs and RAW tools, respectively. The corresponding testing results in terms of confusion matrices are shown in Fig. 3-8. By comparing these results based on eigen demosaicing features with those in Table 2-3 and Table 2-5 based on SFFS-selected demosaicing features, we find that our eigen demosaicing features generally lead to better accuracies with a smaller feature size. For identification of the fourteen DSC models in Table 2-2 using the same PSVM classifier, our 20 eigen demosaicing features achieve an average identification accuracy of 98.0% while the SFFS-selected 250 demosaicing features achieves 97.5% accuracy. For identification of the ten RAW tools in Table 2-4, our 20 eigen demosaicing features achieve an average accuracy of 99.7% while the SFFS-selected 50 demosaicing features achieve 99.1% accuracy. Since SFFS feature selection usually requires very long time especially when the feature pool is large, ERE also requires significantly less time in the feature reduction process.

Though SFFS feature selection is inferior to ERE in many aspects for our forensics identification tasks, we note that feature selection, e.g. using SFFS, and subspace feature reduction, e.g. using ERE, are two types of fundamentally different methodologies. In a test scenario, feature selection has the advantage that only the selected features need to be computed while our full demosaicing feature set still need to be computed for subspace feature reduction methods. Therefore, feature selection technique can save the feature extraction time. In practice, feature selection techniques can also be used in conjunction with subspace feature reduction techniques to improve the trade-off between feature extraction efficiency and the accuracy of the forensics analyzer.

3.3.5 Identification of Same-Model DSLR Cameras

In this experiment, we test the goodness of using our eigen demosaicing features to identify the individual DSLR cameras of the same model. With 1000 images from five different Canon EOS 5D cameras and based on a similar experimental setup, our 20 eigen demosaicing features yield an average accuracy of 52.5% to identify the five copies of Canon EOS 5D cameras. With 800 images from four different Canon EOS 40D cameras, our 25 eigen demosaicing features yield an average accuracy of 54.0% to identify four copies of Canon EOS 40D cameras. Though still much better than the accuracies corresponding to the random-guess scenarios, these poor accuracies, as compared with our 98.0% accuracy in identifying fourteen DSC models in Fig. 3-8(a), justify our claim that our demosaicing features are more appropriate and reliable to identify different camera models with different processing pipelines than the individual cameras of a same model, where each individual camera is expectedly to have almost identical processing pipeline.

3.4 Summary

In this chapter, we propose to combine three types of demosaicing features for identification of mobile camera models. Comparison results for the nine-cam-model identification shows that our combined demosaicing features achieve an average test accuracy of 99%, which confidently outperforms several conventional statistical image forensics features and their suggested combinations. This shows that our proposed demosaicing features are an excellent choice for identification of the low-end mobile cameras from their output images. By including numerous cameras of the same model and very close models, in fifteen-cam identification, our average test accuracy in model identification of the individual cameras is still as high as 94.8%. In such a case, we find from the obtained confusion matrix that to some extent, the cameras of the same model expectedly tend to mix with each other. Moreover, by extending ERE feature reduction method to DSC model and RAW-tool identification, we find that better identification accuracies are achieved with a reduced feature size as compared with the SFFS feature selection technique in our Chapter 2.

Chapter 4 Ensemble Tampering Detection on Image Patches Using FusionBoost and Demosaicing Features

In this chapter, we formulate the universal image tampering detection as a large asymmetrical binary pattern classification problem and propose a solution framework using a novel ensemble fusion procedure, called FusionBoost, together with accurately detected demosaicing features. We first extract and divide the large set of image demosaicing features into small feature subsets according to both the demosaiced sample category and the feature type. For each feature subset, we train a lightweight tampering detector using probabilistic support vector machine. FusionBoost is then proposed to learn the weights of an ensemble tampering detector for achieving the minimum error rate. By applying the ensemble tampering detector on cropped photo blocks from different sources, large-scale experiments show that our proposed framework performs extremely well in detecting a variety of common image tampering operations. Comparison results show that FusionBoost performs better than the conventional ensemble learning and classifier fusion techniques. In an application of patch-based tampering detection examples, we show the efficacy of our proposal in detecting local image tampering.

4.1 Introduction

As digital photos are still widely and continuously used as important visual evidences in growing number of applications such as journalist reporting and police investigation, reliable photo authentication and tampering discovery are paramount to restore the public trust towards the digital visionary. The prior works on passive forensics tampering detection generally fall into 2 categories, tampering specific and universal. The tampering specific approaches detect the specific anomaly leftover by a particular tampering operation, e.g. region duplication [67, 72, 81, 93, 98], resampling [72, 76], splicing [69-70, 84, 97] and double JPEG recompression [72, 78, 83, 88, 91]. While the universal approaches typically detect the disturbances and inconsistencies on some intrinsic and fragile image regularities, which can be caused by a number of tampering operations or their combinations. The image regularities used in previous forensics works include lighting consistency [74, 86, 87], sensor noise pattern [80], demosaicing regularity [82, 90], high-order wavelet statistics [77] and image quality metrics [68, 77], etc. These regularities are either bounded by physics law or formed in the camera processing pipelines and detection of the different regularities work well for different forensics scenarios.

The objective of this research is to simultaneously and reliably detect a wide range of common image tampering operations so that a forensics analyst can analyze a selected suspicious local image patch to determine whether it is original from a source as claimed or it has been tampered afterwards. Since many types of image tampering exist, the space of tampered images is huge as compared with the genuine untampered image space for a given source. Hence, we formulate the detection problem as a large asymmetrical binary pattern classification task. In the proposed solution, we use accurately detected demosaicing features from different demosaiced sample categories for classification. Our syntactic study shows that the extracted demosaicing features automatically contain the distortion signatures of various image tampering; hence can be effectively used to identify various post-manipulations. By first dividing the overall demosaicing features into small feature subsets, we train a number of lightweight tampering detectors using probabilistic support vector machine (PSVM). Through FusionBoost learning, the individual classifiers are linearly combined into a strong

ensemble classifier by repetitively selecting a new classifier from a pool of individual tampering detectors and adaptively assigning the classifier weight. The key contribution of our proposed system include: 1) We propose a two-step learning framework, which simplify a highly asymmetrical learning task with diversified high-dimensional features into two easy-to-manage steps, i.e. learning a set of light-weight PSVM tampering detectors and fusion with FusionBoost learning. Both steps address the asymmetry problem with suitable solutions provided; 2) We introduce a new set of accurately detected demosaicing features for the task of universal image tampering detection. Experimentally, our features are shown to be highly effective; 3) A novel FusionBoost learning is proposed to combine a pre-trained set of probabilistic tampering detectors. Comparison results show the strength of FusionBoost over the conventional fusion techniques and ensemble learning methods. Besides, we propose to normalize the individual probabilistic tampering detectors to operate at the equal error rate (EER) threshold of 0.5 for probabilistic scores using an exponential function. We have also considered different tampering attempts on a large number of images used in our experiment. The scale of our experiment is large as compared with several previous works [77, 82, 90]. Experimental results show that our ensemble classifier works extremely well in tampering detection.

In addition, the proposed framework offers a salient advantage for passive image forensics. To make deliberate image forgery harder, it has been advocated in [67, 109] that comprehensive image forensics should be based on a suite of forensics tools, which examine different image aspects. This literally means combining different forensics experts and diversified forensics features. Such concept is well supported in the proposed framework, which allows diversified forensics experts to be combined into a strong ensemble forensics expert. The probabilistic score of the ensemble detector can be used as the confidence score, which helps a human forensics analyst to decide whether a local image region has been tampered.

The remainder of this Chapter is organized as follows. Sec. 4.2 details the proposed method. Sec. 4.3 demonstrates experimentally the effectiveness of our ensemble tampering detector and the FusionBoost learning. Sec. 4.4 summarizes this chapter.

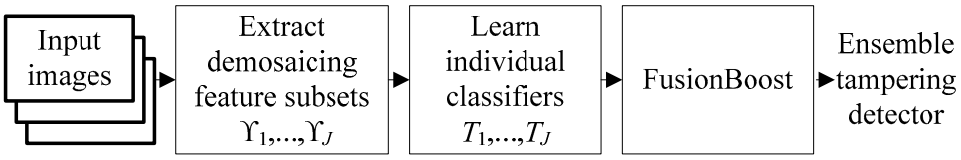


Fig. 4-1 Flow Graph in Construction of Ensemble Tampering Detector

4.2 Proposed Tampering Detection Framework

Similar to the object detection tasks in literature, detection of image tampering can be formulated as an asymmetrical binary learning problem, which classifies a given image either into the untampered class or the tampered class. Since many different forms of tampering exist, a representative tampered class is usually huge as compared to the untampered class. Fig. 4-1 shows the flow graph of the proposed construction of the ensemble tampering detector. Given a set of training images, we first extract demosaicing features for each image and train a set of lightweight probabilistic tampering detectors using different demosaicing feature subsets. FusionBoost is then used to learn the weights of each individual classifier in order to construct a strong ensemble tampering detector. In the following section, we describe the techniques in details.

4.2.1 Demosaicing Features

Our extraction of demosaicing features is the same as what is described in Chapter 2. Based on partial derivative correlation models, we extract a total of 1248 weights (WT), 256 error cumulants (EC) and 32 normalized group sizes (NGS) for 16 different demosaicing sample categories. In both Chapter 2 and 3, we have demonstrated that these features are highly effective in discriminating various common image sources including digital still cameras, RAW-tools and mobile cameras. In this work, we further verify their effectiveness in detection of common image manipulations. We first collect a syntactic set of 200 color images containing a large variety of scenery. After re-demosaicing these images with Hamilton's algorithm [135], we separately apply 13 types of manipulations related to brightness adjustment, contrast adjustment, sharpening, various image filtering, resampling and lossy JPEG compression. We analyze how the

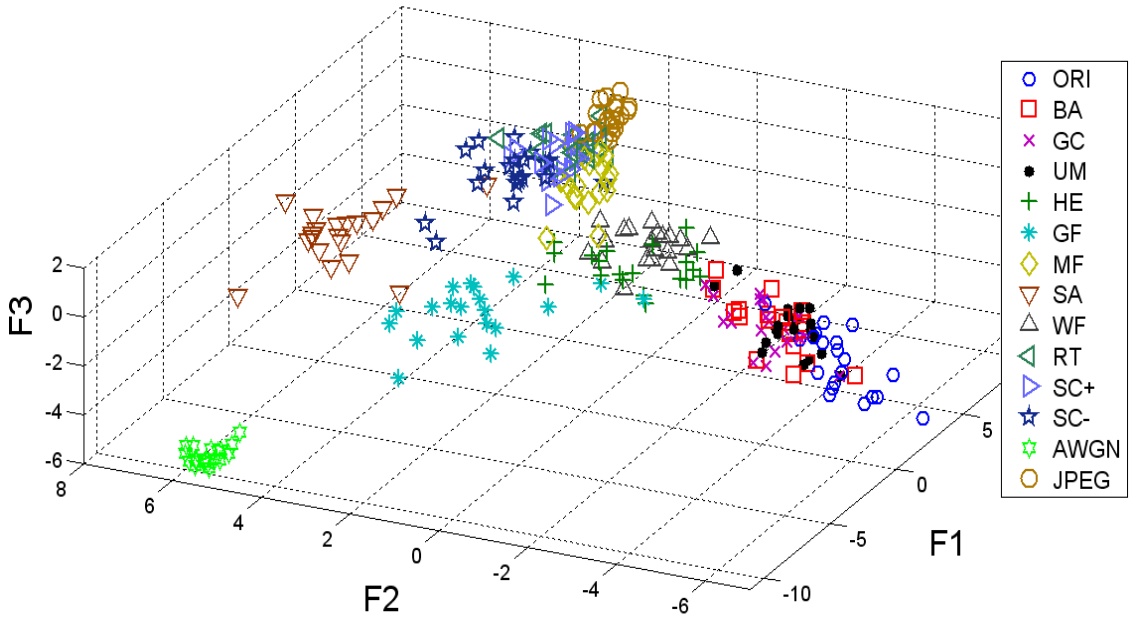


Fig. 4-2 When Projected on a 3D Linear Discriminant Feature Subspace, the Original Images, which are Demosaiced by Hamilton's Algorithm [135], and its 13 Types of Manipulated Images Form Clusters

demosaicing features are distorted by the common image manipulations. By letting the original images be one class and each of the 13 tampered types to be another class, we form a total of 14 classes. Principal component analysis (PCA) is then performed to reduce the feature dimension to 30, followed by linear discriminant analysis (LDA). The projections onto the 3 most discriminant LDA axes are shown in Fig. 4-2, where image samples from the same class tend to cluster together. The acronyms of the 14 classes in Fig. 4-2 are: ORI: untampered images; BRI: brightness enhancement by 10 gray levels of for an 8-bit representation; GC: gamma correction with $1/\gamma=1/1.6$; UM: unsharp mask with 30 dB distortion; HE: histogram equalization; GF: Gaussian filtering with $\sigma=1$; MF: median filtering with 3×3 window; SA: special averaging with 3×3 window; WF: Wiener filtering with 3×3 window; RT: counter clockwise rotation by 5°; SC+: scaling Up by 1.1x; SC-: Scaling Down by 0.9x; AWGN: Additive White Gaussian Noise (PSNR =30 dB); JPEG: Lossy JPEG Compression with a Quality Factor of 80. Even in a 3D subspace, we find, in many cases, the original class is separable from majority of the different manipulated classes. By utilizing the 30 PCA features together with support vector machine classification, we find that the original class can be separated from each of the 13 manipulation classes with nearly 100% accuracy.

This good result shows that the different manipulations easily distort the underlying demosaicing regularity and leave distinctive traces, which are being captured reliably into our demosaicing features. This makes the demosaicing features possible good choices in detecting post-manipulations. However, demosaicing is usually not the last process in an image creation pipeline and many different demosaicing algorithms exist. The effectiveness of using the demosaicing features in tampering detection needs further investigation on real photos from diversified sources.

4.2.2 Learning Individual Tampering Detector

In order to build the lightweight individual tampering detectors, we divide the extracted demosaicing features into 20 feature subsets $\Upsilon_1,\ldots,\Upsilon_{20}$ according to both the demosaicing feature type and the demosaiced sample category. Shown in Table 4.1 are descriptions of these feature subsets. Each subset contains the features extracted based on the four different Bayer CFAs in Fig. 2-1(d). Since the feature subsets are either extracted from different demosaiced sample categories or from different feature types, diversified performances are expected for detecting different tampering.

With each feature subset, we train an individual universal tampering detector using PSVM with radial basis function (RBF) kernel, which is implemented in the LIBSVM tool [147]. Support vector machine (SVM) is a powerful nonlinear pattern classification tool. By minimizing the structural risk [117] and utilizing the kernel trick, SVM typically exhibits good generalization performance with the unseen data. The works in [118, 119] further extend the traditional SVM into probabilistic SVM (or PSVM) by studying distribution of the SVM outputs and map them into probabilistic scores by optimizing a sigmoid function. In our study, we find that PSVM generally performs better than SVM and more importantly, the probabilistic outputs from different individual classifiers can be better combined to construct a stronger ensemble classifier. It should be noted that for a large asymmetrical learning task, we find it difficult to achieve good result by directly following the LIBSVM guild [146] in the classifier training. Instead, we make the following changes:

1) The feature vectors extracted from our original class are repeated many times

Table 4-1 Twenty Demosaicing Feature Subsets

Feature Set	Type	Demosaiced Sample Categories			Dimension
		Color	Sensor Color	Axis	
Υ_1	WT	G	r	*x*	52
Υ_2				*y*	52
Υ_3				*o*	104
Υ_4			b	*x*	52
Υ_5				*y*	52
Υ_6				*o*	104
Υ_7		R	g	*x*	104
Υ_8				*y*	104
Υ_9			b	*u*	52
Υ_{10}				*v*	52
Υ_{11}				*o*	104
Υ_{12}		B	g	*x*	104
Υ_{13}				*y*	104
Υ_{14}			r	*u*	52
Υ_{15}				*v*	52
Υ_{16}				*o*	104
Υ_{17}	EC	G	All	All	96
Υ_{18}		R	All	All	80
Υ_{19}		B	All	All	80
Υ_{20}	NGS	ALL	All	All	32

Note: The word 'All' represents all applicable sub-categories. In the "Axis" column, *x*: vertical axis; *y*: horizontal axis; *u*: minor diagonal axis, *v*: main diagonal axis and *o*: omnidirectional or the average axis; To interpret the feature sets, for instance, the Υ_1 feature set includes all 52 weights (WT) features from the category of green (G) demosaiced samples located on the red (r) sensor samples with *x* as the predetermined demosaicing axis. Similarly, description of the various demosaiced sample categories can be found in Fig. 2.5

with additive random noises doped to make sizes of the untampered and the tampered classes comparable before the data are fed into PSVM training. This boosts the prior probability of the untampered class and expands the decision boundary towards the tampered class. Experimentally, we find that by adding a small amount random noise, the equal error rate (EER) measured on the cross-validation test data usually improves for the PSVM classifier. This can be explained that some good support vectors are created among the noisy training feature vectors and these make the PSVM classification boundary learned during the training have better tolerance on the feature measurement errors. In our study, we let the additive noise be uniformly distributed in an empirically determined range of $[-0.05, +0.05]$ after each feature is linearly scaled into the range of [-1, +1];

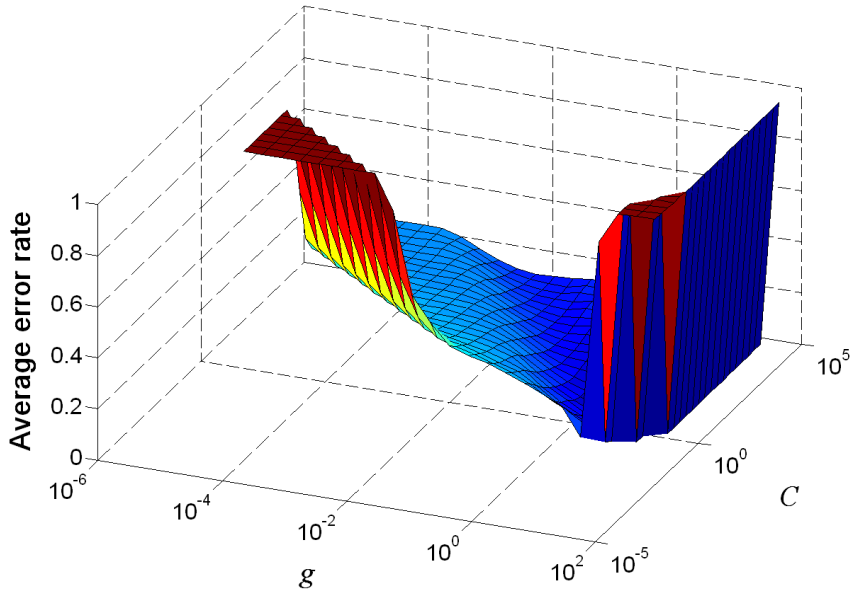


Fig. 4-3 Log-Scale Grid Search of the Best Parameters for an Individual PSVM Tampering Detector with the (C, g) Corresponding to the Lowest Average Error Rate Being Selected. Average Error Rate Here Refers to the Average of Type-I and Type-II Error Rates

2) Performance of a PSVM with RBF kernel is largely dependent on parameters (C, g) [146]. As shown in Fig. 4-3, we perform a log-scale grid search to find the best parameter combination, which minimizes the average of type-I and type-II error rates on a cross-validation feature set. Here, the type-I and type-II error rates are respectively the error rate of classifying an untampered image as the tampered class and the error rate of classifying a tampered image as the untampered class. Instead, the LIBSVM guild [146] suggests performing log-scale grid search of (C, g) based on the criterion of minimizing the total error rate, i.e. the number of errors divided by the total number of images. In our tampering detection task, this criterion tends to bias towards the class with a larger size and generally does not work well for our asymmetric learning task, where size of the tampered class is significantly larger than the untampered class.

With the best parameters determined for the j^{th} feature subset, we perform cross validation (CV) to generate the CV probabilities for the j^{th} tampering detector $T_j(x)$, where x denotes an input image. For a v-fold CV, we first randomly divide the training samples into v equal partitions. Each time, we select one test partition and the remaining $v-1$ partitions are used for training. The trained PSVM tampering detector is then applied to the test partition to have the corresponding CV probabilities. After choosing

each of the v partitions as a test partition, we gather a large set of CV probabilities $\{p_{nj}, y_n\}$ for $T_j(x)$, where $p_{nj} \approx T_j(x_n) = P(y_n = 1 | x_n, T_j)$, $y_n \in \{0,1\}$ is the class label with '1' denoting the untampered class and '0' denoting the tampered class, $1 \le j \le 20$ and $1 \le n \le N$, where N denotes the number of available training images. Note that we can choose a large v so that the statistics of our CV probabilities best match with those measured with the unseen data. However, since a larger v would require more training time, we select v=10.

With the CV probabilities, we measure the equal error rate (EER) and its corresponding EER threshold for each tampering detector. We find that the measured EER thresholds $\{t_j\}$ are inconsistent for the tampering detectors $\{T_j(x)\}$ associated with different feature subsets $\{\Upsilon_j\}$. Their values can differ significantly from the natural threshold of 0.5 for probabilistic scores. Since it is desirable to operate each tampering detector on a common basis so that they can be better combined at a later time, we perform the following nonlinear normalization

$$f(p) = p^A \tag{4.1}$$

where p and $f(p)$ are respectively the original and the normalized probabilistic score and $A \neq 0$ is an exponential parameter. The normalization in Eqn (4.1) automatically satisfies $f(0) = 0$ and $f(1) = 1$. We further require $f(t) = \frac{1}{2}$ so that

$$t^A = \frac{1}{2} \tag{4.2}$$

where t denotes the currently measured EER threshold of the tampering detector $T(x)$ and $\frac{1}{2}$ is the desired EER threshold. By taking logarithm and dividing the term $\log(t)$ on both side of Eqn (4.2), we derive

$$A = -\frac{\log(2)}{\log(t)} \tag{4.3}$$

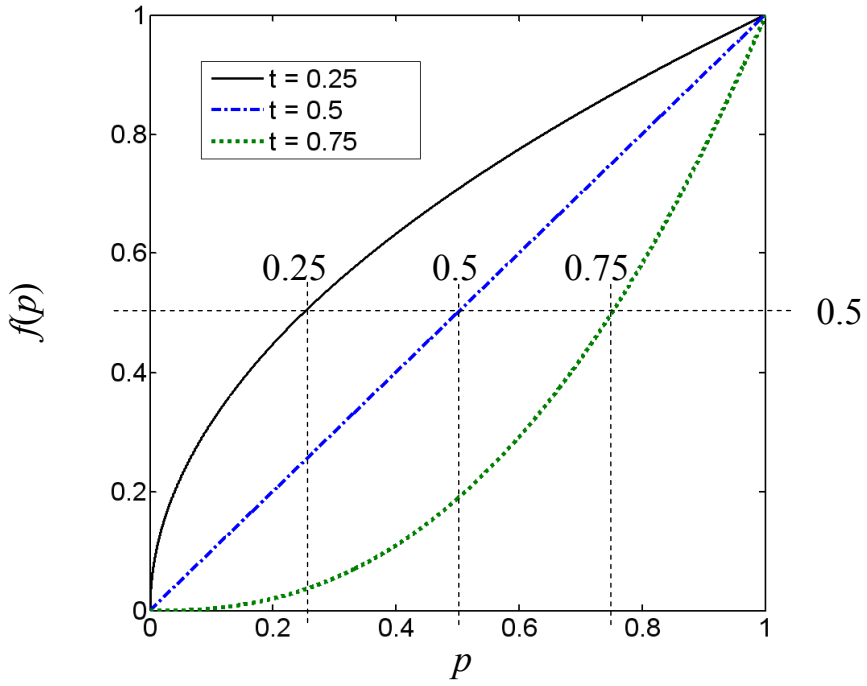


Fig. 4-4 Normalization Curves for Probabilistic Scores Corresponding to Different EER Thresholds

Shown in Fig. 4-4 are several normalization curves corresponding to different EER thresholds t. These normalization curves have the desired properties of being smooth and monotonically increasing, which ensure that 0.5, the natural threshold of probabilistic scores, is the new EER threshold after the normalization. In the following section, we let $\{T_j(x)\}$ represent the normalized individual tampering detectors.

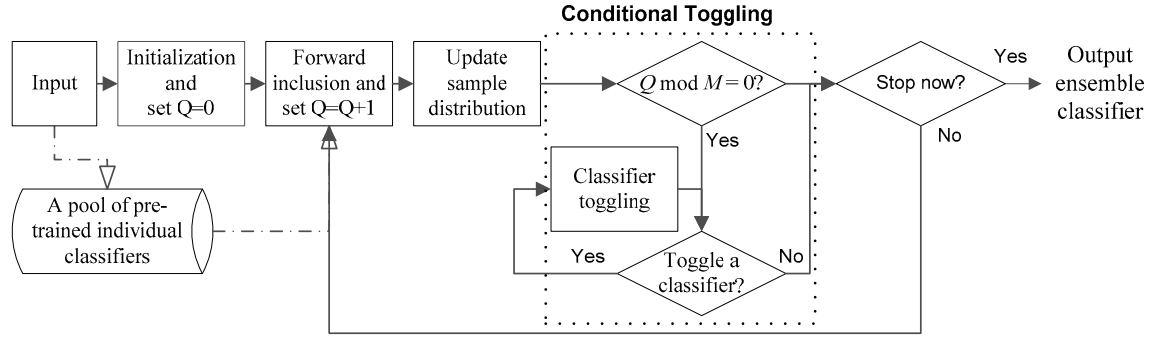
4.2.3 Constructing Ensemble Tampering Detector

In this section, we linearly combine the pool of individual tampering detectors with the ensemble probabilistic classifier written as

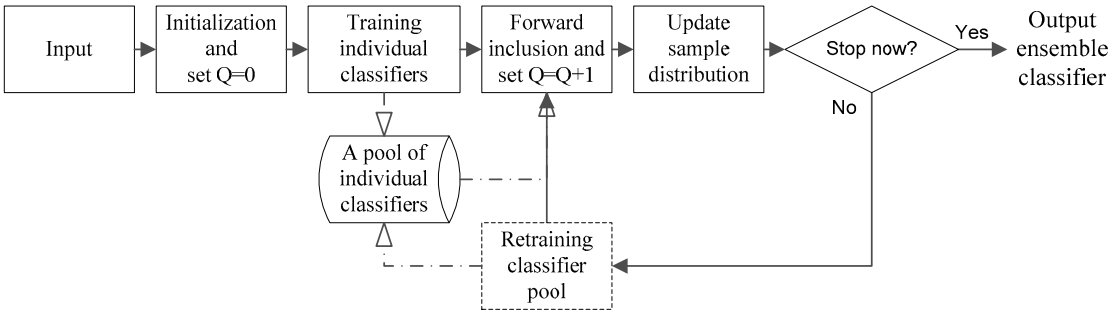
$$T(x) = \sum_{j=1}^{J} a_j T_j(x) \tag{4.4}$$

where $\{a_j\}$ for $1 \le j \le J$ denote the weights and they satisfy

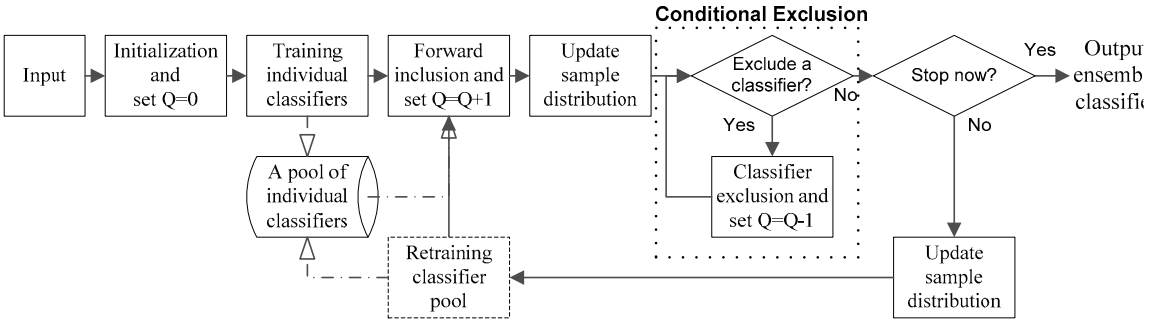
$$\sum_{j=1}^{J} a_j = 1$$



(a) FusionBoost



(b) RealBoost



(c) FloatBoost

Fig. 4-5 Comparison of Flow Graph for the Proposed FusionBoost, RealBoost [121] and FloatBoost [122]

Since the different individual tampering detectors are trained using different feature subsets, diversified classification performances are expected. At the same time, we also expect some individual classifiers to have correlated performances. It is important to properly determine the weights $\{a_j\}$ in order to achieve good performances. In this study, we propose a FusionBoost learning to iteratively find the best weights. The flow graph of FusionBoost is shown in Fig. 4-5 (a) in comparison with the flow graphs of RealBoost [121] and FloatBoost [122] in Fig. 4-6 (b) and (c) respectively. It should be

noted that FusionBoost is leveraged on the development of RealBoost with a specific aim of combining a pre-trained set of probabilistic classifiers. The details of RealBoost for two classes are described in Appendix A. Similar to the FloatBoost, FusionBoost incorporates a floating search based backtracking mechanism, but in a different manner. In the following section, we present FusionBoost algorithm in details with its main differences and comparative strengths highlighted.

Fig. 4-6 shows the detailed procedures of FusionBoost. Like other boosting algorithms, FusionBoost has the intrinsic capability to handle asymmetrical learning tasks. In step 1, the initialization, the initial total weights distributed to the untampered class and to the tampered class are made equal to reflect their equal importance in the learning objective. Since there are more samples in tampered class, after equal weight division within a class, the weight associated with each tampered sample is significantly smaller than that for a sample in the untampered class.

In the FusionBoost, we can write our stagewise ensemble probabilistic classifier as

$$S_Q\left(x,\kappa_1^{(Q)},\ldots,\kappa_Q^{(Q)}\right)=\frac{\sum_{q=1}^{Q}\kappa_q^{(Q)}\omega_q s_q\left(x\right)}{\sum_{q=1}^{Q}\kappa_q^{(Q)}\omega_q} \tag{4.5}$$

where $s_q\left(x\right)$ is the q^{th} selected classifier from the pre-trained classifier pool $\{T_j\left(x\right)\}$, Q is the current number of classifier stages, $\kappa_q^{(Q)}\in\{0,1\}$ is a control flag, ω_q is the weight associated with $s_q\left(x\right)$. The output class label is written as $R\left(S_Q\left(x,\kappa_1^{(Q)},\ldots,\kappa_Q^{(Q)}\right)\right)$, where $R\left(p\right)$ rounds a real-valued probabilistic score p to its nearest integer. An error occurs if $\left|y-S_Q\left(x,\kappa_1^{(Q)},\ldots,\kappa_Q^{(Q)}\right)\right|\geq 0.5$, where $y\in\{0,1\}$ denote the class label. By maintaining and updating a sample distribution, RealBoost [121] iteratively trains and selects a new classifier with adaptive weight assignment. The iterative process minimizes the upper bound of an exponential loss function (See Eqn (B.9) in Appendix B) and optimizes the separation margins by always emphasizing more on the harder samples. In step 2 of FusionBoost, we follow the RealBoost framework and modify it for our probabilistic classifiers. The details of converting RealBoost for

1. Input and initialization

(1) Given labeled training sample images $(x_1, y_1),\ldots,(x_n, y_n),\ldots,(x_N, y_N)$ where $y_n = 0,1$ are the labels for the tampered and the original samples respectively.

(2) Given a pool of individual classifiers $T_1(x),\ldots,T_j(x),\ldots,T_J(x)$ and the normalized cross-validation probabilistic scores $\{p_{nj}\}$, where $1 \le n \le N$, $1 \le j \le J$ and $p_{nj} = T_j(x_n)$.

(3) Initialize $D_n^{(0)} = \frac{1}{2\ell}, \frac{1}{2(N-\ell)}$ for $y_n = 0,1$ respectively, where ℓ is the number of tampered samples and $\{D_1^{(Q)},\ldots,D_N^{(Q)}\}$ is the sample weight distribution for the Q^{th} stage;

(4) Initialize $Q = 0$ and $\xi(S_Q) = \xi\left(S_Q\left(x, \kappa_1^{(Q)},\ldots,\kappa_Q^{(Q)}\right)\right) = 1$, where $\xi(S_Q)$ denotes the average of type-I and type-II error rates for an ensemble classifier S_Q.

2. Forward Inclusion

(1) Set $Q = Q+1$ and $\kappa_Q^{(Q)} = 1$. Set $\kappa_q^{(Q)} = \kappa_q^{(Q-1)}$ for $1 \le q \le Q-1$;

(2) Select each $T_j(x)$ as $s_Q(x)$, find the weight ω_Q of $s_Q(x)$ using Eqn (4.6) and measure $\xi\left(S_Q\left(x, \kappa_1^{(Q)} = 1,\ldots,\kappa_Q^{(Q)} = 1\right)\right)$, the average of Type-I and Type II error rates for the ensemble classifier $S_Q\left(x, \kappa_1^{(Q)} = 1,\ldots,\kappa_Q^{(Q)} = 1\right)$.

(3) Choose the $s_Q(x)$ from $\{T_j(x)\}$, which gives the minimum average error rate $\xi\left(S_Q\left(x, \kappa_1^{(Q)} = 1,\ldots,\kappa_Q^{(Q)} = 1\right)\right)$. Update ω_Q using Eqn (4.6);

(4) Update the sample distribution $D_n^{(Q)}$ using Eqn (4.7) for $1 \le n \le N$ and linearly normalize $\{D_n^{(Q)}\}$ so that $\sum_n D_n^{(Q)} = 1$;

(5) Update $\Omega_Q = \Omega_{Q-1} \cup s_Q$ and compute $\xi(S_Q)$. $\Omega_Q = \{s_1,\ldots s_Q\}$ are the selected individual classifiers and $\Omega_0 = \varnothing$ is initialized in the step 1;

3. Conditional toggling

(1) If $Q \bmod M = 0$, where M is period of conditional toggling, then

(a) Let $\xi_q(S_Q) = \xi\left(S_Q\left(x, \kappa_1^{(Q)},\ldots,\kappa_{q-1}^{(Q)}, \left|\kappa_q^{(Q)} - 1\right|, \kappa_{q+1}^{(Q)},\ldots,\kappa_Q^{(Q)}\right)\right)$ corresponds to the error rate, where the flag $\kappa_q^{(Q)}$ is toggled its opposite state. Find $i = \arg\min_{1 \le q \le Q} \xi_q(S_Q)$;

(b) If $\xi_i(S_Q) < \xi(S_Q)$, then

i) Set $\kappa_q^{(Q)} = \left|\kappa_q^{(Q)} - 1\right|$ and update $\xi(S_Q)$;

ii) goto step 3.(1)(a);

(c) else goto step 2.(1)

(2) else goto step 2.(1)

4. Output

(1) With a good number of classifier stages included, i.e. Q, find $I = \arg\min_{1 \le q \le Q}\left(\xi\left(S_Q\right)\right)$;

(2) Construct $T(x)$ from $S_I\left(x, \kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)$ using Eqn (4.4) and Eqn (4.10).

Fig. 4-6 FusionBoost Learning Procedures

combining our probabilistic classifiers are described in Appendix B. Given the current ensemble classifier S_{Q-1}, $s_Q(x)$ is selected to minimize the average error rate

$\xi\left(S_Q\left(x,\kappa_1^{(Q)}=1,\ldots,\kappa_Q^{(Q)}=1\right)\right)$. The weight ω_Q associated with $s_Q(x)$ is derived as (See Eqn (C.12) and Eqn (C.13) in Appendix C)

$$\omega_Q=\frac{1}{2}\ln\left(\frac{\varepsilon+\sum_{\forall|y-s_Q(x)|<0.5}D_n^{(Q-1)}}{\varepsilon+\sum_{\forall|y-s_Q(x)|>0.5}D_n^{(Q-1)}}\right) \tag{4.6}$$

where ε is a small smoothing constant, say 10^{-9}, and the new sample distribution $\{D_n^{(Q)}\}$ is updated by (See Eqn (B.11) in Appendix B)

$$D_n^{(Q)}=D_n^{(Q-1)}\times\exp\left(2\omega_t\left|y_n-s_Q\left(x_n\right)\right|\right) \tag{4.7}$$

To achieve better performance, we introduce in step 3 of our FusionBoost a backtracking mechanism, called conditional toggling, to switch off the unfavorable stages. As in Fig. 4-5, the comparison of FusionBoost and the conventional RealBoost and FloatBoost in the flow graphs, we highlight that conditional toggling is the main difference between FusionBoost and two conventional boosting algorithms. It should be noted that including a new classifier stage in RealBoost does not necessarily lead to better error rate. Hence, unfavorable stages exist among all the included stages. FloatBoost [122] introduces a backtracking mechanism, called conditional exclusion, to remove the unfavorable classifier stages every time when a new classifier stage is included. It is claimed in [122] that FloatBoost typically results in fewer classifier stages with comparable performance of RealBoost. Compared with the conditional exclusion in FloatBoost, which removes the unfavorable stages every time a new stage is included, FusionBoost has the following differences in the backtrack design:

1. FusionBoost toggles the on/off control flags of the included stages instead of removing an unfavorable stage. This allows the *off*-ed stages to be switched *on* again if it helps with performance in the future iterations.
2. The backtracking in FusionBoost is designed to be completely independent from the forward inclusion. This is done by temporally resetting the control flags of $\{\kappa_1^{(Q)},\ldots,\kappa_q^{(Q)},\ldots,\kappa_Q^{(Q)}\}$ to be a default of "1" in our forward

inclusion steps 2.2 and 2.3 in Fig. 4-6.

3. A period M is empirically selected to control frequency of the backtracking. By selecting M=1, the highest backtracking frequency, flow graph of FusionBoost in Fig. 4-5(a) would degenerate to be similar to that of FloatBoost in Fig. 4-5(c) in the backtracking portion except that FusionBoost uses conditional toggling. If we select $M = \infty$, the backtracking in FusionBoost is disabled and its flow graph would degenerate to be similar to RealBoost in Fig. 4-5 (b). In both extreme cases, we find that the performance of the FusionBoost is no better than a properly selected M.

Besides these differences in the backtracking design, FusionBoost adopts a more objective criterion in selecting the next classifier, i.e. to minimize the average of type-I and type-II error rate. It should also be noted in Fig. 4-5 that, as an optional step, re-training the classifier pool based on the updated sample distribution is implicitly supported by both RealBoost and FloatBoost. However, re-training a large classifier pool is a highly computationally intensive procedure and requires simple and special individual classifiers, which can take the distribution $\{D_n^{(Q)}\}$ as an input. It is also inapplicable to forensics fusion scenarios that some given forensics classifiers act as black boxes, which do not allow re-training. Therefore, as far as our concern in this work, FusionBoost combines a set of pre-trained probabilistic classifiers.

By deactivating the retraining blocks for RealBoost and FloatBoost in Fig. 4-5 and based on our earlier obtained CV probabilities, we find experimentally that RealBoost, which does not backtrack, frequently achieves better error rate than FloatBoost if a large number of iterations are provided. The inferior result of FloatBoost is likely caused by its frequent backtracking and update of the sample distribution. Consider a possible scenario that adding two stages improves the current ensemble but adding the first stage degrades its performance. Highly likely, FloatBoost would remove the first stage once it is included into the ensemble and restores the sample distribution back to its earlier state. This makes FloatBoost easy to be trapped into a deadlock once it hits a local minimum of the error function. The backtracking design in FusionBoost takes into account such issues, hence, usually achieves better performance than RealBoost. Since we do not consider retraining the classifier pool in our fusion problem, the sample distribution need

not be updated to reflect the immediate alterations made in the backtracking. Empirically, we find that the most frequent conditional toggling, which corresponds to a period of M=1, does not give a better tampering detection performance than a relatively larger M. This will be further discussed in our following experimental result section.

The iterative learning in FusionBoost stops after a large number of iterations, say 1000 for a pool of J=20 classifiers. In step 4, FusionBoost finds I, the number of iterations that gives the lowest average error rate. According to Eqn (4.5), $S_I\left(x,\kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)$ can be written as

$$S_I\left(x,\kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)=\frac{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q s_q\left(x\right)}{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q} \tag{4.8}$$

To convert the ensemble classifier $S_I\left(x,\kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)$ into $T\left(x\right)$ in Eqn (4.4), we first substitute $s_q\left(x\right)=\sum_{j=1}^{J}\mu_{jq}T_j\left(x\right)$, where $\mu_{jq}=\begin{cases}1, & \text{if } s_q\left(x\right)=T_j\left(x\right)\\ 0, & \text{otherwise}\end{cases}$, into Eqn (4.8). The ensemble classifier $S_I\left(x,\kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)$ can be re-expressed as

$$\begin{aligned}S_I\left(x,\kappa_1^{(I)},\ldots,\kappa_I^{(I)}\right)&=\frac{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q\sum_{j=1}^{J}\mu_{jq}T_j\left(x\right)}{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q}\\&=\frac{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q\mu_{1q}}{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q}T_1\left(x\right)+\ldots+\frac{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q\mu_{Jq}}{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q}T_J\left(x\right)\end{aligned} \tag{4.9}$$

By comparing Eqn (4.9) and Eqn (4.4), the weights $\{a_j\}$ in Eqn (4.4) can be written as

$$a_j=\frac{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q\mu_{jq}}{\sum_{q=1}^{I}\kappa_q^{(I)}\omega_q} \tag{4.10}$$

4.3 Experimental Results and Discussion

As mentioned earlier, the proposed solution is evaluated to classify the following

Table 4-2 The Different Photo Sources Included in the Experiment

Camera Models		RAW Converted Using RAW Tools	
1	Canon Ixus i	7	ACDSee v10
2	Canon 400D	8	Capture One v3.7.8
3	Nikon D70	9	Corel PaintShop Pro 12
4	Olympus E500	10	Olympus Master v2
5	Sony Alpha 350	11	Photoshop CS2
6	Sony P73	12	Picture Window Pro v4

classes,

$\boldsymbol{C}_0$: an image is original from a given source as claimed. The source can be a particular camera, a RAW conversion tool or potentially a group of similar image processing pipelines, e.g. different cameras of the same model;

$\boldsymbol{C}_1$: an image is not the direct output from the given source as claimed, where some tampering takes place or the image is from a different source.

In a practical scenario, we can search in the photo's *exchangeable image file format* (EXIF) headers for information about the 'claimed' source. EXIF headers are popularly available for photos in JPEG and TIFF formats. As a common practice, camera manufacturers store the photo-related information such as the source, shot settings and the time that picture is taken, etc. into the EXIF headers of each photo. If a photo is manipulated and resaved into the same JPEG or TIFF format, the EXIF header data are commonly preserved by the image editing tools. Though EXIF headers can be easily modified or removed, it is difficult to re-establish the harmony between the image content and the EXIF headers once it is destroyed.

To prepare simulation image datasets for $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$, we first select 200 photos from each of the twelve different sources in Table 4-2. These photos cover a large variety of common scenery under different lighting conditions. For the six camera models, the photos are stored in the default JPEG format and the RAW-converted photos in TIFF format. For the RAW photos, the original RAWs are captured by the same Olympus E500 camera. The photo sizes vary from one source to anther with the smallest of 2274×1704 for the Canon Ixus camera and the largest of 4592×3056 for the Sony Alpha350 camera. To have reliable results, we increase the experiment scale by cropping

Table 4-3 Tampering Types and Operations Included in the Experiment. In the Last Column, "4×" Refers to 4 Times of the Size of the Original Untampered Class

Tampering types	Operations	Remarks	Number of Photos
Luminance-plane manipulations	Brightness adjustment (BA)	{+5, +10, +20, +30}	4x
	Gamma correction (GC)	$1/\gamma$ = {1/0.6, 1/0.8, 1/1.3, 1/1.6}	4x
	Unsharp mask (UM)	PSNR = {10dB, 20dB, 30dB, 50dB}	4x
	Histogram equalization (HE)	-	1x
Filtering	Gaussian filtering (GF)	σ = {1, 1.6}	2x
	Median filtering (MF)	Filter order = {3, 7}	2x
	Spatial average (SA)	Filter order = {3, 5, 7, 9}	4x
	Wiener filtering (WF)	Filter order = {3, 7}	2x
Resampling	Rotation (RT)	Degree = {5°, 10°, 15°, 20°}	4x
	Scaling (SCA)	{0.5x, 0.7x,0.9x, 1.1x, 1.3x, 1.5x}	6x
Additive noise	Additive white Gaussian noise (AWGN)	PSNR = {5dB, 10dB, 20dB, 30dB}	4x
Lossy compression	JPEG	Quality factor = {99, 95, 90, 80, 70, 60}	6x
Shifting	One-pixel shift (OPS)	Direction = {→, ↓, → and ↓ }	3x
Photomontage	Reliable sources (RS)	Refer to Table III for details	11x
	Internet sources (IS)	*Flickr* photos from different sources	5x
Total			62x

three non-overlapping blocks of 512×512 along the central row of each photo. This triples the number of images per source to 600. In our simulation, we test the ensemble tampering detection for several diversified sources, which are separately used as the ground-truth sources for class $\boldsymbol{C}_0$.

To construct a tampered image set for class $\boldsymbol{C}_1$, we consider a wide range of tampering types as tabulated in Table 4-3. These include various luminance-plane manipulations, filtering, resampling, additive noise, one-pixel shifting and the photomontage, which involves mixing image signals from other sources. The tampered images, except for photomontage, are generated by applying the corresponding manipulation to the original images for $\boldsymbol{C}_0$. For photomontage, we include the following two different types of other sources. The reliable sources refer to the remaining eleven sources in Table 4-2, which are not used for $\boldsymbol{C}_0$. The Internet sources contain 1000 downloaded photos from the *Flickr* website. The photo sources span 18 major camera brands and over 75 different camera models. The dimensions of these photos are in comparable range to those from reliable sources. The number of images for $\boldsymbol{C}_1$ is 62 times of those used for $\boldsymbol{C}_0$. The total number of images used in each of our tampering detection experiment is 37800.

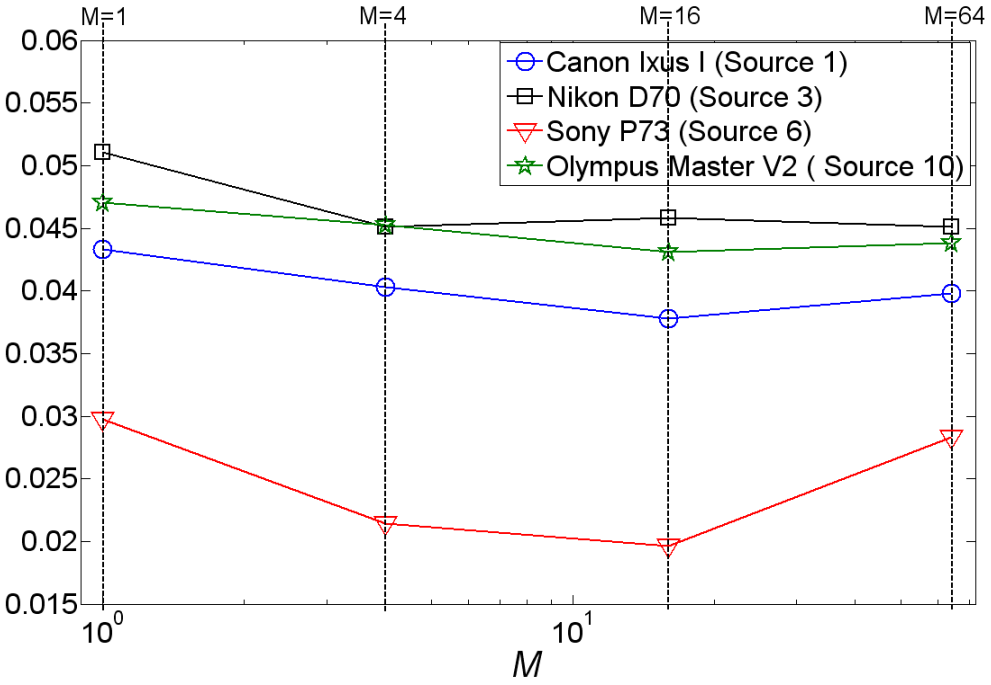


Fig. 4-7 Average Tampering Detection Error Rate versus the Period M of Conditional Toggling in the FusionBoost Learning

4.3.1 Tampering Detection Experiment

By randomly selecting the cropped images from about 3/4 of the photos for training and the remaining for testing, we extract the demosaicing features and construct the ensemble tampering detector. As an example, we conduct the tampering detection experiment on 4 selected diversified sources in Table 4-2 including Canon Ixus I (source 1), Nikon D70 (source 3), Sony P73 (source 6) and Olympus Master v2 (source 10). Shown in Fig. 4-7 is the average of Type-I and Type II error rates at different backtracking periods for the four different sources. From the results, we can see that M=16 leads to very good tampering detection results, which are consistently better than the results of the most frequent backtracking case, i.e. M=1. Hence, we set M=16 for our following tampering detection experiments.

Table 4-4 Detection Error Rate (%) for Canon Ixus Camera. Refer to Table 4-2 for Acronyms of the Tampering Types. T-1 and T-2 Error Rates Represent the Type-I and Type-II Error Rates Respectively

		Individual Tampering Detector																				Ensemble
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}	detector T
T-1 Error Rate		19	16	46	22	20	36	9	11	20	16	14	7	11	23	22	10	14	21	17	21	5
T-2 Error Rate	BA	60	71	48	62	75	52	13	14	68	69	80	12	21	73	77	85	81	57	41	62	8
	GC	53	63	46	55	72	45	12	14	62	62	74	9	18	68	71	77	79	51	36	54	7
	UM	37	38	37	32	44	32	11	11	51	53	42	12	17	61	56	46	53	34	21	34	7
	HE	12	17	33	13	34	14	2	7	45	42	22	1	3	50	41	56	46	0	1	22	0
	GF	0	1	11	1	1	1	0	0	6	2	5	0	0	2	1	6	33	0	10	9	0
	MF	10	5	10	3	1	1	0	0	7	13	0	0	0	17	14	0	17	0	2	5	1
	SA	0	0	6	0	0	1	0	0	3	4	0	0	0	5	3	0	22	0	8	5	0
	WF	30	37	33	28	32	19	0	1	56	51	3	3	1	40	44	9	29	1	16	11	0
	RT	1	3	19	1	0	1	0	0	2	6	1	2	0	7	5	1	21	0	0	25	0
	SCA	0	3	18	1	1	2	0	0	6	12	2	1	0	5	6	1	28	0	2	26	0
	AWGN	2	5	4	7	0	1	0	0	0	0	0	0	0	2	1	1	2	0	0	5	0
	JPEG	40	34	37	19	31	30	4	3	53	46	27	0	0	66	51	26	1	0	0	48	0
	OPS	10	10	18	12	10	12	11	1	10	16	4	16	4	14	14	4	9	0	1	23	0
	RS	5	3	10	4	11	9	6	6	16	10	3	7	5	15	12	5	54	7	6	17	6
	IS	10	6	13	11	19	19	20	9	18	21	10	18	12	22	20	12	41	11	14	27	8

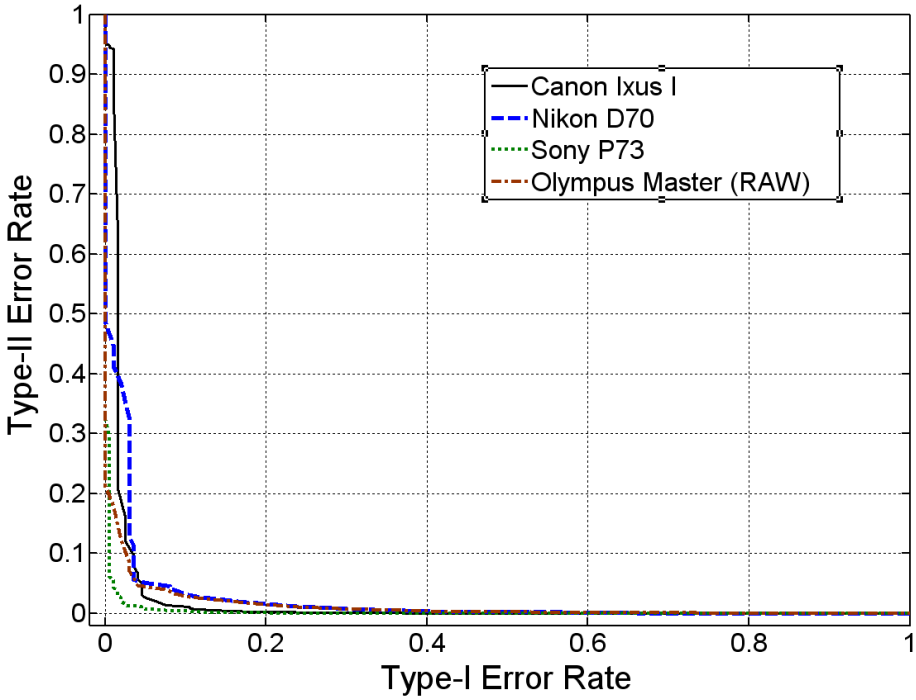


Fig. 4-8 Receiver Operating Characteristics for the Ensemble Tamper Detectors Constructed for Different Image Sources

Shown in Table 4-4 are the detailed testing error rates for the Canon Ixus camera. The best performing tampering detectors are $T_7(x)$ and $T_{12}(x)$, which are trained respectively using feature subsets Υ_7 and Υ_{12}. According to Table 4-1, these feature

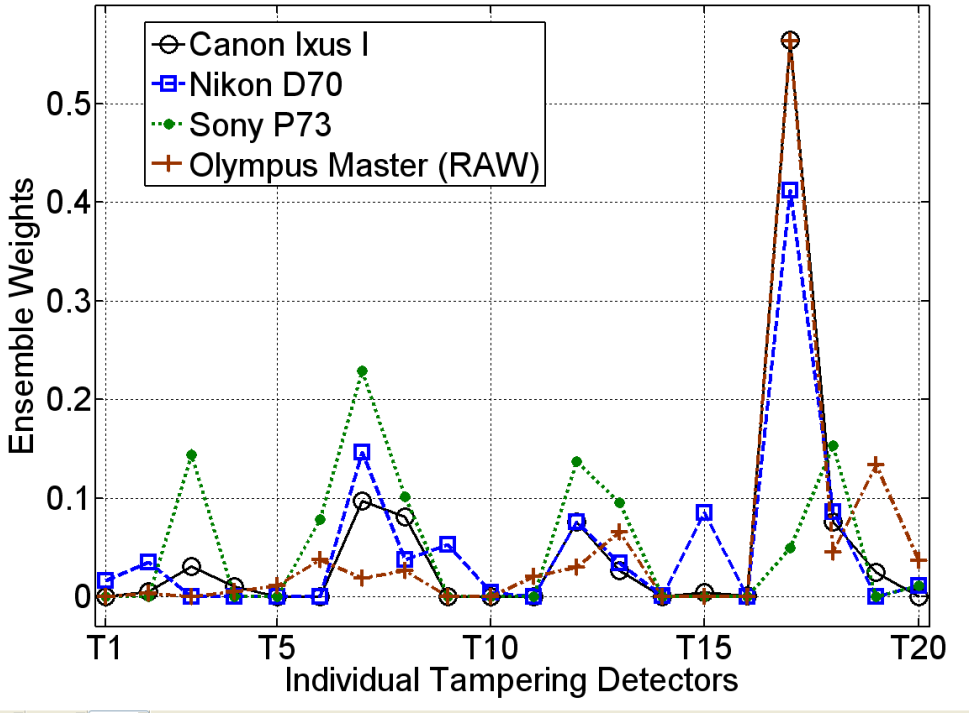


Fig. 4-9 The Weight Profile Learned by FusionBoost for the Ensemble Tampering Detectors Constructed for Different Image Sources

Table 4-5 Comparison for RealBoost, FloatBoost and Proposed FusionBoost in Terms of Average Test Error Rate (%) for Ensemble Tampering Detection on Different Original Image Sources. Refer to Table 4-2 for the Image Sources

Image Sources (#)	RealBoost	FloatBoost	FusionBoost
1	4.2	4.5	3.8
3	4.5	4.8	4.2
6	3.1	2.9	2.0
10	4.6	5.6	4.3

subsets contain the *WT* features extracted from the red and blue demosaiced sample categories respectively with *x*-axis being the demosaicing axis. The average error rate of 3.8% is achieved by the ensemble detector, which is significantly lower than 7%, the best error rate achieved by the individual tampering detectors. This shows that FusionBoost has well utilized the relatively weaker individual classifiers to construct a stronger ensemble classifier.

Among the different tampering types, we find that our ensemble detector works

extremely well for filtering, resampling, additive noise, lossy JPEG compression and one-pixel shifting. Such operations likely introduce more distinctive distortions to the underlying demosaicing regularity. Though the original images from the Canon Ixus camera are in JPEG format, JPEG re-compressing these images again with various quality factors can still be detected reliably. This is due to the large differences in JPEG implementations, particularly in chroma sub-sampling and the different quantization tables used. Our detection error rate for the luminance-plane operations such as brightness adjustment, gamma correction and sharpening is about 7-8%. Considering such processes are likely implemented as a camera post-demosaicing processes, the achieved error rate is satisfactory. The error rates achieved for detecting images from other sources is as low as 6% and 8%, respectively, for the reliable sources and the Internet sources. The small error rates suggest photomontage forgery is largely detectable.

For other $\boldsymbol{C}_0$ sources including Nikon D70, Sony P73 and Olympus Master RAW tool, our constructed ensemble tampering detectors achieve similar average test error rate of 4.2%, 2.0% and 4.3%, respectively. These good results collectively demonstrate the effectiveness of our proposed tampering detection framework for the photos from diversified sources. Shown in Fig. 4-8 and 4-9 are, respectively, the receiver operating characteristic curves and the ensemble weight profiles. From the weight profiles, we find the largest weight is commonly associated with $T_{17}(x)$, which is trained using the green-channel EC features. Though $T_{17}(x)$ does not give highly accurate result by itself, its large weight suggests it well and uniquely complements the rest individual tampering detectors. We also note that about 7-11 individual tampering detectors have zero weight associated. These tampering detectors can be readily removed from the classifier ensemble and the corresponding demosaicing features need not be computed. This reduces the system complexity.

4.3.2 Comparison with Other Fusion Algorithms

Table 4-5 is the comparison of RealBoost, FloatBoost and FusionBoost in combining the 20 probabilistic tampering detectors. Though, for all the different sources, FloatBoost stops within only a few iterations, it gives the largest error rate. The proposed

Table 4-6 Test Error Rate (%) Achieved by the Classical Fusion Method Including Mean, Product (Pro.), Minimum (Min.), Median (Med.), Majority Vote (M.V.) and Feature-Level Fusion (FLF). Refer to Table 4.2 for the Image Sources

Image source #	Combining All 20 Individual Tampering Detectors					Combining the Best 5 Individual Tampering Detectors					FLF
	Mean	Pro.	Min.	Med.	M.V.	Mean	Pro.	Min.	Med.	M.V.	
1	8.1	8.6	8.8	9.8	10.6	4.4	6.1	6.6	4.5	6.2	4.2
3	10.2	9.4	8.5	11.9	11.9	5.6	5.4	5.5	6.3	6.5	6.5
6	8.2	19.1	18.8	9.3	9.4	4.7	5.0	5.0	4.6	5.1	4.2
10	9.0	15.4	15.9	8.0	8.0	6.5	9.1	10.5	6.5	8.9	8.2

FusionBoost consistently gives the lowest error rate. The average number of iterations required for FusionBoost is about 120, which is smaller than 320, the average number of iterations for RealBoost. Due to the backtracking mechanism, the required time for FusionBoost to complete the same number of iterations is about 4 times of RealBoost.

Besides boosting algorithms, we have tested several other traditional fusion strategies including feature level fusion and classifier combination using different rules [148]. The results are tabulated in Table 4-6. Compared with them, our FusionBoost result in Table 4-5 is clearly better.

4.3.3 Application to Patch-Based Tampering Detection

In a practical scenario, a composite of different tampering types is commonly used to create a forgery photo. Since our ensemble tampering detector simultaneously detects many tampering types, it can be used for generic image tampering detection. In this section, we test its efficacy in discovering local image tampering in a patch-based manner. With a patch size of 512×512 and a step size of 256, we scan through an entire photo step by step along both the horizontal and the vertical axes. For each local block, we extract the demosaicing features, feed the feature subsets into the corresponding individual tampering detectors and combine the output probabilities into an ensemble probabilistic score. Since each small area of 256×256 can be traversed up to 4 times with different ensemble scores, we let the average score to indicate the confidence of authenticity for that region.

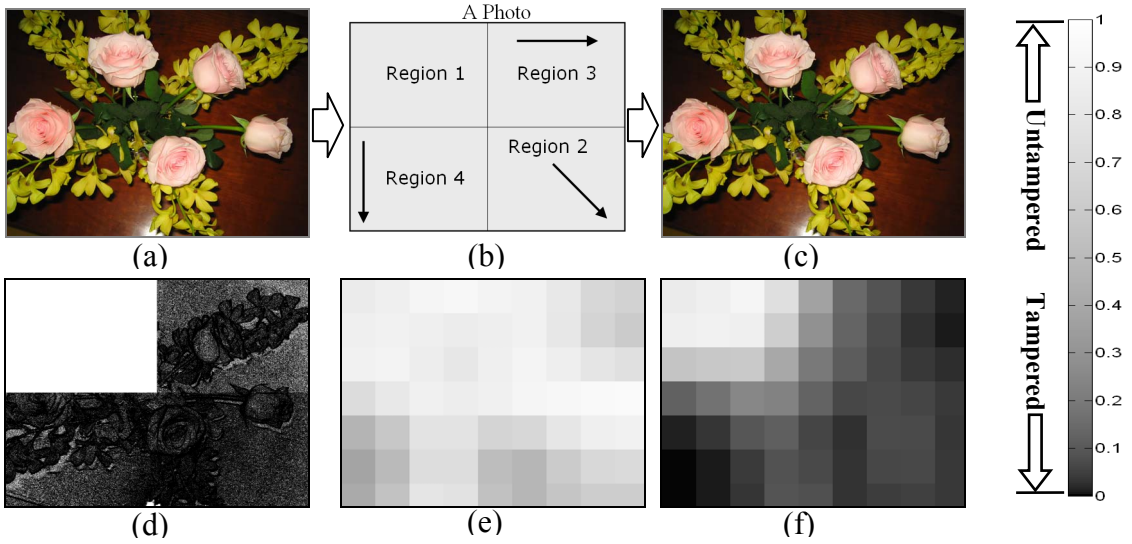


Fig. 4-10 Patch-Based Tampering Detection Using FusionBoost-Learned Ensemble Tampering Detector on a Region-Shift Forgery; (a) An Original Photo Taken by Canon Ixus i Camera; (b) Illustration of Creating One-Pixel Region-Shift Forgery; (c) Region-Shifted Version of (a); (d) the Difference Map Between (c) and (a), Where the Region in Black Color Denotes At Least One Gray-Level Difference for an 8-Bit Representation Per Color; (e), (f) Block-Based Tampering Detection Probability Maps on Photos in (a) and (c) Respectively

Fig. 4-10 and Fig. 4-11 show two forgery detection examples for the Canon Ixus camera. Since the demosaicing regularity is believed to be consistent across an entire original photo, we use the previously trained ensemble detector with the central-region image blocks to discover the possible tampering in the entire photo. Shown in detection result are the average scores for each small area of 256×256. In the first example, we detect the region-shift forgery. As illustrated in Fig. 4-10(b), the region-shift forgery is created by shifting the Region 2 to 4 by one pixel in the respective three arrow directions. Though the small modifications are hardly noticeable by human eyes, the one-pixel shifts change the underlying CFA correspondingly to its three shifted versions. Since our extracted demosaicing features contain the discriminative CFA information, such tampering can be detected effectively. From the detection results in Fig. 4-10(e) and (f), we clearly see that the shifted regions have very low confidence scores as compared to the detection result on the original image. In the second example, the forgery photo in Fig. 4-11(b) is created from Fig. 4-11(a) using a number of tampering operations such as resampling, region cloning, blur filtering, etc. The detection result in Fig. 4-11(e) shows the tampered region is largely detectable. In both examples, we find

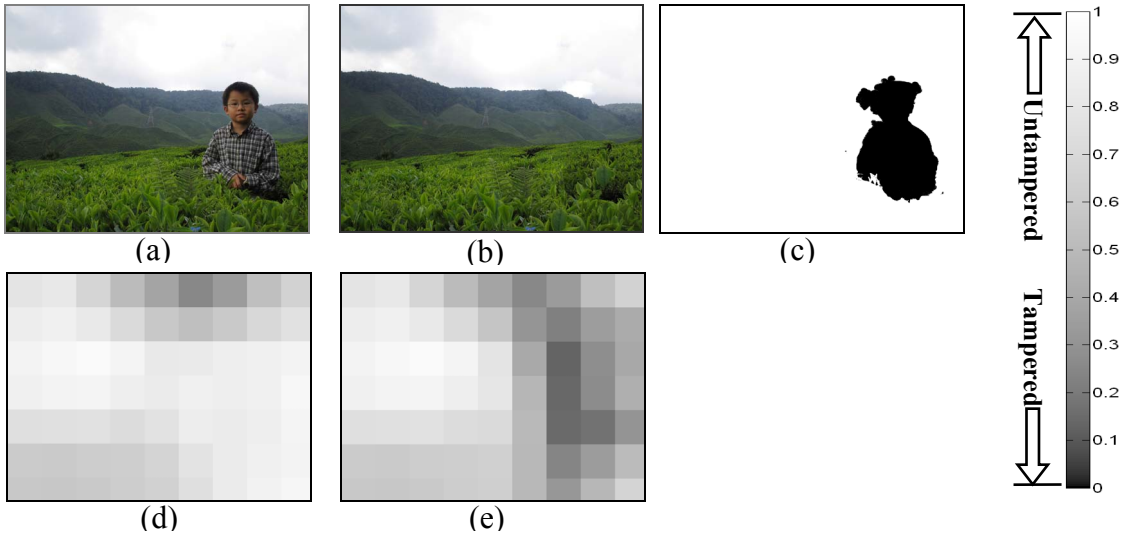


Fig. 4-11 Patch-Based Tampering Detection Using FusionBoost-Learned Ensemble Tampering Detector on a Common Object-Removal Forgery; (a) An Untampered Photo Taken by Canon Ixus I; (b) the Tampered Version of (a); (c) the Difference Map Between (a) and (b); (d), (e) Block-Based Tampering Detection Probability Maps on Photos in (a) and (b) Respectively

that the detection on the two original images work better in the central regions though the results achieved in the edge regions are also acceptable. This suggests that there are still some statistical differences among our extracted demosaicing features at different image regions, which can be caused by factors like richness of image content or different sensor noises. To further improve the result, it is recommended to include the training images cropped at the other regions as well. The detection result in Fig. 4-11(d) shows some obvious false alarms in the sky region because it is generally difficult to extract high-quality demosaicing regularity in the smooth and saturated image regions as we have discussed in Chapter 2.

4.4 Summary

In this chapter, we propose a universal tampering detection framework to simultaneously detect a wide range of tampering types. By dividing demosaicing features into a number of small feature subsets, we learn a set of lightweight tampering detectors using probabilistic support vector machine. These individual classifiers exhibit diversified classification performances. After linearly combining them into an ensemble

classifier using FusionBoost, large-scale experiments show that our ensemble detector achieves very low average error rates ranging from 2.0%-4.3% to detect tampering for different original image sources. The ensemble detector works particularly well to detect various image filtering, resampling, additive noise, lossy JPEG compression and one-pixel shift. Though luminance domain operations and photomontage are relatively more difficult to detect, our results of less than 8% are still highly satisfactory. Comparison results also show that our proposed FusionBoost is highly effective in constructing a classifier ensemble. Consistent better performances in combining 20 individual probabilistic classifiers are achieved by FusionBoost than by other ensemble learning approaches and the conventional classifier fusion strategies. Moreover, the classifier selection in FusionBoost eliminates the ineffective or redundant tampering detectors and the corresponding feature subsets, which reduces the system complexity.

Chapter 5 Identification of Recaptured Images on LCD Screens

With advances in the image display technology, recapturing good-quality images from the high-fidelity artificial scenery on a LCD screen becomes possible. Such image recapturing posts a security threat, which allows the forgery images to bypass the current forensics systems. In this chapter, we first recapture some good-quality photos on different LCD screens by properly setting up the recapturing environment and tuning the controllable settings. In a perceptional study, we find that such carefully recaptured photos can hardly be identified by human eyes. To prevent this image recapturing threat, we propose a set of statistical features, which capture the common anomalies introduced in the image recapturing process on LCD screens. With a probabilistic support vector machine classifier, comparison results show that our proposed features work extremely well and outperform the conventional image forensics features in identification of the carefully recaptured images.

5.1 Introduction

To restore the trustworthiness of digital images, image forgery detection [111] has been intensively studied in recent years through detection of certain intrinsic image regularities or some tampering anomalies. Frequently, the tell-tale cues useful for image forensics such as lens distortion, sensor noise pattern and statistics, demosaicing regularity and JPEG characteristics are directly associated with the image creation pipeline, where the light signals are converted into a digital image. Though some

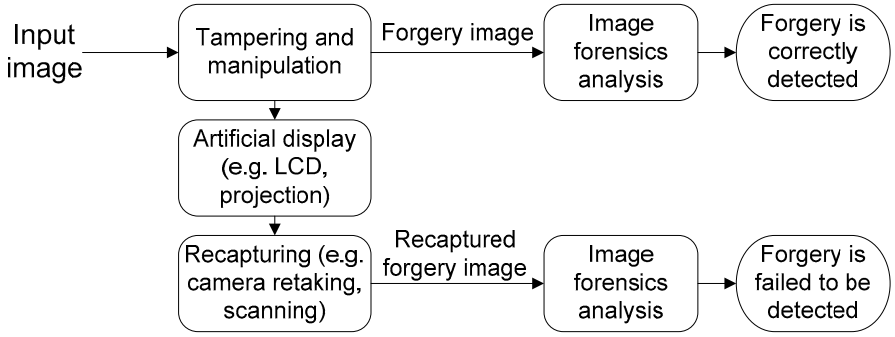


Fig. 5-1 Comparison of Image Forensics Results on the Direct Forgery Images and the Recaptured Forgery Images

forensics methods can efficiently expose the direct tampering made on an image, most existing methods are unable to expose the indirect scenery forgery, where the scenery to be captured is artificially created. Creating a physical scenery in general can be a very difficult and expensive task. With the aid of today's ubiquitous and high-fidelity display technology, generating a virtual scenery of reasonable fidelity is still relatively easy and such technology is potentially exploited to defeat the current image forensics systems.

As illustrated in Fig. 5-1, an image forger can first display the tampered images with a high-quality artificial display e.g. through high-fidelity printing, liquid crystal display (LCD) or projection. Through proper set-up, the forger can recapture the artificially generated scenery and use the recaptured image to fool the image forensics system. It should be noted that during the image recapturing process, the tampering anomalies, e.g. splicing discontinuity and resampling artifacts, are automatically removed and the intrinsic image regularities which are originally disturbed due to the tampering operations are automatically restored. Moreover, creating such kind of forgery requires no specialty and can be implemented by novices of modern computer and photography technology. In a real-life example [149, 150], a hunter recaptured a fake tiger scenery, which is made of a paper tiger poster, to prove the very presence of a commonly believed extinct tiger species. After being accepted by the local authority and published on Internet, these recaptured photos sparked large-scale controversy. To prevent such a security loophole, we consider identification of the recaptured images an important task to facilitate the current image forensics systems.

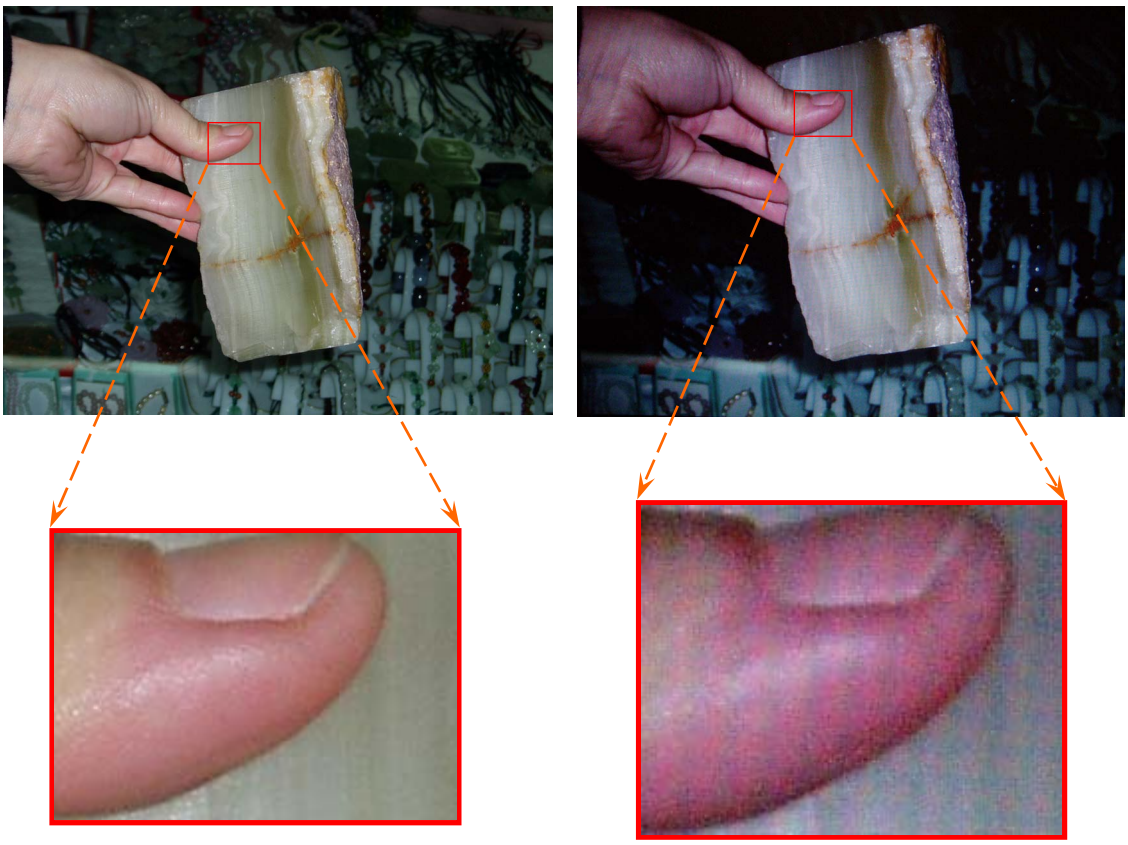
In previous research, the work in [64] initially brought up the issue of "rebroadcast attack" in image forensics. By formulating it as a binary classification problem, the author shows that the printed-and-scanned photos can be accurately distinguished from the natural photos by using 72 wavelet statistical features and a simple linear discriminant classifier. In [63], a set of image geometry features are proposed to distinguish photos from computer graphics (CG). At the same time, the authors show experimentally that both the geometry features and wavelet features [62, 64] can be used to classify recaptured PRCGs on LCD screen from real PRCGs with a good accuracy. In [106], by assuming the artificial scenery is planar, e.g. a face image, the authors find that the specular component measured on recaptured images can serve as a distinctive feature to differentiate a recaptured photo from a natural photo. In this Chapter, we consider identifying carefully recaptured photos on common LCD screens. Since image recapturing is commonly accompanied with some image quality losses, we first study the settings for recapturing good-quality images. A perceptional study is then designed to test the human ability in identifying the carefully recaptured images. Based on our observation, we further propose several sets of image features, including texture features, loss-of-the-details features and color features to identify the recaptured images from natural images.

This chapter is organized as follows: Section 5.2 investigates on the environmental setup and the settings for recapturing good-quality images. Section 5.3 describes a survey on human's ability to identify the finely recaptured images. Section 5.4 proposes several types of image features for automatic identification of recaptured images and experimentally demonstrates their effectiveness. Finally, Section 5.5 summarizes this chapter.

5.2 Recapturing Good-Quality Images

5.2.1 Image Recapturing Artifacts

Casually recapturing a scenery displayed on a LCD screen often lead to poor quality of recaptured images. As shown in Fig. 5-2, we can easily observe some obvious



(a) Natural image (b) Recaptured image

Fig. 5-2 Comparison of a Natural Image and its Casually Recaptured Version on a HP NC4000 Notebook Computer's LCD Screen, Where the Same Olympus U300d Camera is Used for Capturing Both Images

artifacts such as the texture patterns, loss of fine details and color degradation from the poor-quality example.

When enlarged in Fig. 5-2(b), the texture patterns on the causally recaptured image are clearly visible. These patterns are closely related with the artifacts of LCD display panel. The entire LCD display panel typically consists of a large number of tiny liquid crystal display cells and each cell further contains the separate red, green and blue display units. Normally, these tiny cells cannot be singled out by human eyes unless some defective cells are present, e.g. the always lit or unlit cells. The LCD cells also dynamically change due to periodical charging and re-charging at a high rate, which is beyond the frequency response range of human eyes. Periodical inverse polarization driving method [151] is usually implemented on common LCDs to avoid the image sticking problems. Though the tiny structures and the dynamic changes are hardly

perceptible to human eyes, these artifacts can still be sensed by the camera sensor and manifest as tiny regular textured patterns.

The loss-of-the-details artifacts are also clearly seen such as the blurred object edges and unidentifiable small detailed structures, e.g. the grains on the finger skin in Fig. 5-2 (b). Two possible reasons contribute to this. Most cameras have installed an anti-aliasing filter to blur the scenery in order to make sure that the very sharp edges on the scenery do not cause the aliasing effects. Since the recaptured photo has gone through this blurring filter more than once, the resultant image experience stronger aggregate blurring. The second reason is related to the supported resolution of the commercial LCD monitors. Currently, the commonly supported maximal resolution is about 1280×1024 for a 19-inch LCD monitor, which is only slightly greater than a one-megapixel image. However, existing commercial digital still cameras can easily support up to ten-megapixel photos. To display a higher resolution photo on a low-resolution LCD, this photo need be down-sampled to fit the LCD resolution. Hence, significant information loss occurs even before the photo is recaptured.

Some obvious color degradations are also commonly observable on a recaptured image. Many commercial cameras seem to be unable to correctly sense the color temperature of the LCD-displayed images. Hence, the white point detected is error-prone and the resultant images tend to appear bluish. Another artifact is that the four corners of a retaken photo tend to be darker than the central regions, which is analogous to the vignetting artifacts of the traditional photography.

5.2.2 Setting for Good-Quality Recapturing

Generally, due to the poor visual quality, the casually recaptured photos are useless and they can be easily identified by human eyes. The question now becomes whether we can recapture reasonably good-quality images from the ubiquitous LCD screens in a common environment so that they can be used to fool human eyes.

To perform such a test, we have set up an image recapturing environment as illustrated in Fig. 5-3. It should be noted that such an environment contains a large

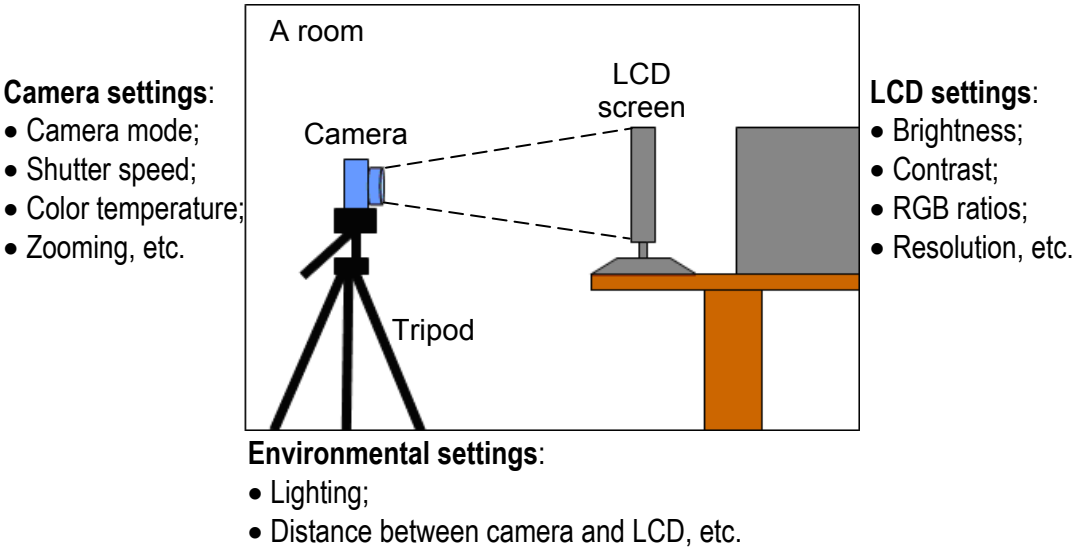


Fig. 5-3 Our Image Recapturing Environment Where the Tunable Settings to Achieve Better Quality of Recaptured Images are Highlighted

number of controllable settings including the camera settings, the LCD settings and the environmental settings. These settings can be tuned to recapture good-quality images.

With three available digital still cameras (Canon Powershot A620, Olympus Mju 300 and Olympus E500 DSLR) and three common LCDs (Philips 19" 190B6CG, NEC 17" AccuSync and Acer 17" AL712), we experimentally determine the best settings for each of the nine camera-LCD combinations by systematically tuning all the controllable settings. The best determined setting differs reasonably from one combination to another. Generally in the tuning process, we found that the following settings tend to give better visual quality of the recaptured images:

- *Camera settings:*

  Camera mode: Shutter speed priority mode is preferred as this mode allows us controlling the exposure time directly. For some point-and-shoot cameras without such a mode, night mode is preferred instead;

  Shutter speed (or exposure time): Shutter speed is usually set in the range from 0.3 to 1.3 seconds depending on the brightness of the LCD monitors and their refresh rates. If the shutter speed is too short, the strip patterns will be more obvious.

If the shutter speed is too long, the recaptured photos will look a bit washed out with greater loss of image details. At the optimal shutter speed, we find artifacts like the strip patterns become less noticeable by human eyes and image details are still reasonably preserved;

Color temperature: The color temperature setting determines the parameters to be used for white balancing inside the camera. Usually a low color temperature setting like under water and in house tends to give us better color quality as compared to the original images;

Zooming: We find that large zooming and a far distance between the camera and the LCD tend to give us less obvious strip patterns. Hence, we stick to the maximal allowable zooming in our environmental setup in Fig. 5-3;

- *LCD settings:*

  Brightness: We choose 100% for most cases;

  Contrast: We set it as 60% to make image edges more prominent. Usually the default is 50%;

  RGB ratios: Usually we set it as 1:1:1. For cameras that do not have a suitable white balancing setting, we can adjust these RGB ratios to achieve the desired color mixture on the recaptured photos.

  Resolution: This setting is always set to be the maximally supported resolution of LCD monitors in order to have less loss of image details.

- *Environmental settings:*

  Lighting: A dark room with no light is preferred so that all the lights come from the LCD display. This avoids the unexpected background lighting and reflections;

  Distance between camera and LCD: This distance is set to be about 6 meters, which is the maximal allowable distance in our room setup. The zooming is adjusted

Natural Images Recaptured images

Fig. 5-4 Two Pairs of Nature and Recaptured Images for Training

accordingly to capture the entire LCD screen.

Though with the best setting, we can still observe some obvious quality degradation if we compare the recaptured images with their corresponding original images, visual quality of these finely recaptured images is significantly better than the casually recaptured images.

After determining the best recapturing setting, we recapture 300 photos from LCD-displayed scenery for each camera-LCD combination, where the displayed contents are 300 natural photos. Out of these 300 natural photos, 100 are taken by the three available cameras, 100 are downloaded from *Flickr* website and the rest 100 are tampered photos where about 10% of the total image area has been altered. Three modes of object-based tampering considered for creating the tampered photos include object insertion, deletion and color changes. In such a way, we form a finely recaptured dataset of 2700 photos for a total of nine camera-LCD combinations.

5.3 Human Identification of Recaptured Images

To find out how well human beings can identify these finely recaptured images, we perform the following survey in a trained scenario. Since most our survey participants

have not seen a recaptured image before, we first explain about the recapture images on LCDs and provide each participant with two sample pairs of the natural and the recaptured images as shown in Fig. 5-4. On a computer screen, the participant can closely examine the image differences caused by the recapturing. After about 2 to 5 minutes' training and when the participant feels confident, we let the participants browse through 50 selected photos as shown in Fig. 5-5, which contain 20 natural photos and 30 recaptured photos mixed in random order. The participants are allowed to take their own time to closely examine each image before they classify the image either into the "natural" category or the "recaptured" category.

In planning the above experimental protocol for this subjective test, we mainly considered the following factors: 1) Our objective is to mimic the practical context that the human tester does not have multiple versions of the same image for making a decision. We find it relatively easy to spot out a natural image when it is mixed in a random order with its several recaptured versions by comparing visual quality of the images. But when only one image is present, we find it hard to make a good decision; 2) In deciding to use only two pairs of training images, we consider comfort and representativeness are two most important factors. Since it is impossible to load a human tester with too training images, we have carefully chosen two pairs of representative images of common scenery for the tester to learn the typical image differences caused by recapturing on LCD screens. As we do not impose a time constraint in the training, a human tester can still finish this training within a comfortable period. This user comfort is important to allow the human tester patiently conducting the remainder part of the test, which assures the accuracy in our subjective-test results.

This survey is conducted on 30 participants, who are mostly university students and staffs. By comparing the participants' answers with the ground-truth answers, the detailed survey results are tabulated in Table 5-1. From them, we find on average, the type-I error rate is 19.8%, i.e. natural images been misclassified as "recaptured" and the type-II error rate is 51.1%, i.e. the recaptured images been misclassified as "natural". The standard deviations of the type-I error rate and type-II error rate are respectively 12.8% and 18.3%. These large error rates suggest that common people are poor in differentiating our carefully recaptured photos from the natural photos. Therefore, these

Fig. 5-5 Samples of 50 Images Used in Human Recaptured Image Identification Survey

Table 5-1 Human Classification Survey Results of Natural and Retaken Photos

Participant	T-I errors	T-I error rate	T-II errors	T-II error rate	Overall err	Overall err rate
1	3	15.0%	16	53.3%	19	38.0%
2	2	10.0%	16	53.3%	18	36.0%
3	2	10.0%	18	60.0%	20	40.0%
4	4	20.0%	12	40.0%	16	32.0%
5	2	10.0%	21	70.0%	23	46.0%
6	1	5.0%	21	70.0%	22	44.0%
7	3	15.0%	21	70.0%	24	48.0%
8	5	25.0%	15	50.0%	20	40.0%
9	6	30.0%	18	60.0%	24	48.0%
10	6	30.0%	17	56.7%	23	46.0%
11	5	25.0%	15	50.0%	20	40.0%
12	0	0	21	70.0%	21	42.0%
13	8	40.0%	15	50.0%	23	46.0%
14	0	0	10	33.3%	10	20.0%
15	2	10.0%	11	36.7%	13	26.0%
16	6	30.0%	19	63.3%	25	50.0%
17	4	20.0%	23	76.7%	27	54.0%
18	4	20.0%	6	20.0%	10	20.0%
19	5	25.0%	15	50.0%	20	40.0%
20	2	10.0%	16	53.3%	18	36.0%
21	1	5.0%	12	40.0%	13	26.0%
22	8	40.0%	15	50.0%	23	46.0%
23	1	5.0%	20	66.7%	21	42.0%
24	6	30.0%	20	66.7%	26	52.0%
25	1	5.0%	21	70.0%	22	44.0%
26	7	35.0%	3	10.0%	10	20.0%
27	8	40.0%	6	20.0%	14	28.0%
28	6	20.0%	10	33.3%	16	28.0%
29	9	45.0%	22	73.3%	31	62.0%
30	4	20.0%	5	16.7%	9	18.0%
Average	**3.97**	**19.8%**	**15.33**	**51.1%**	**19.3**	**38.6%**
Std Dev.	**2.57**	**12.8%**	**5.48**	**18.3%**	**5.6**	**11.2%**

T-I error: type-I error, i.e. a natural image been classified as a retaken type

T-II error: type-II error, i.e. a retaken images been classified as a natural image

recaptured images can potentially fool both human eyes and the current image forensics

system.

5.4 Computer Identification of Recaptured Images

5.4.1 Forensics Features

To prevent the security loophole of the image recapturing attack, reliable automatic identification of the finely recaptured images on LCD screens is highly desirable. By formulating the problem as a binary classification task, we propose the following three types of features to capture the unique anomalies introduced by the recapturing process:

Local binary pattern (LBP): As discussed early, texture patterns at fine scale are often easily observable on a poor-quality recaptured image. Though these texture patterns are not obvious in our carefully recaptured image, we consider complete elimination of the textures is very difficult. To capture such texture anomalies, we compute multiple-scale LBP features [152]. The LBP features are a normalized occurrence histogram of some "uniform" patterns computed using the operator $LBP_{P,R}^{riu2}$ [152], where P is the dimension of the angular space and R determines the resolution. The computed LBP features are invariant to image rotation and contrast changes, which are highly desirable properties for our identification of recaptured images. Moreover, by varying R and P correspondingly, LBP features can be easily extended to multiple scales. By first converting an input color image into a gray image, we compute a total of 80 LBP features at multiple scales using the four operators: $LBP_{8,1}^{riu2}$, $LBP_{16,2}^{riu2}$, $LBP_{24,3}^{riu2}$ and $LBP_{24,4}^{riu2}$.

Multi-Scale Wavelet Statistics (MSWS): Loss of fine details is inevitably coupled with the image recapturing on LCD screens. Considering that the amount of information losses can be very different at the fine scale and at the coarse scale, the image details measured at different scales are potentially good features to expose the recapturing forgery. To measure such features, we first perform N-level wavelet decomposition separately on the R, G and B channels using a standard Haar filter. Let {LL, HL, LH, HH}$_{cn}$ represent the n^{th}–level decomposition of the c^{th} color channel, where $1 \le n \le N$ and

Original Image (Canon camera) | Recaptured Image 1 (Canon+Acer LCD) | Recaptured Image 2 (Canon+NEC LCD) | Recaptured Image 3 (Canon+Phillip LCD)

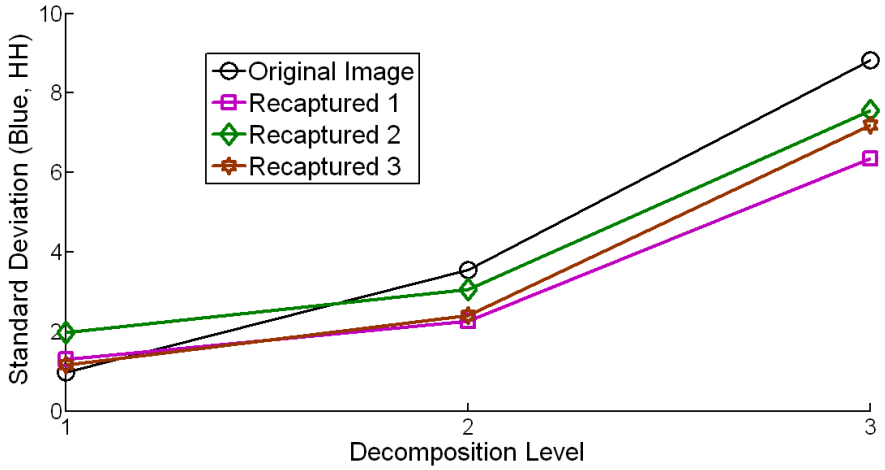


Fig. 5-6 Comparison of the Image-Details Curves at Different Decomposition Levels for an Original Image and Its Three Recaptured Versions

$c \in \{R, G, B\}$. For each high-frequency band in $\{HL, LH, HH\}_{cn}$, we compute the mean and standard deviation of the absolute wavelet coefficients as our features. By setting N=3 and for 3 color channels, we derive a total of 3×3×3×2=54 such features.

Shown in Fig. 5-6 is a comparison of the image details curves at different scales, where the standard deviations of the blue-channel HH sub-bands at different decomposition levels are plotted. From the results, we can observe distinctive changes of characteristic curves due to the recapturing process. At level 1, the standard deviations for the recaptured images are well above that of the original image. It should be noted that though image details have lost significantly at this level, the captured micro-structures usually lead to an increase of the standard deviations measured at this level. At level 2 and 3, the standard deviations of the recaptured images consistently drop below those of the original image. This shows that the loss of image details overweighs the noise contribution from the captured micro-structures at these two decomposition

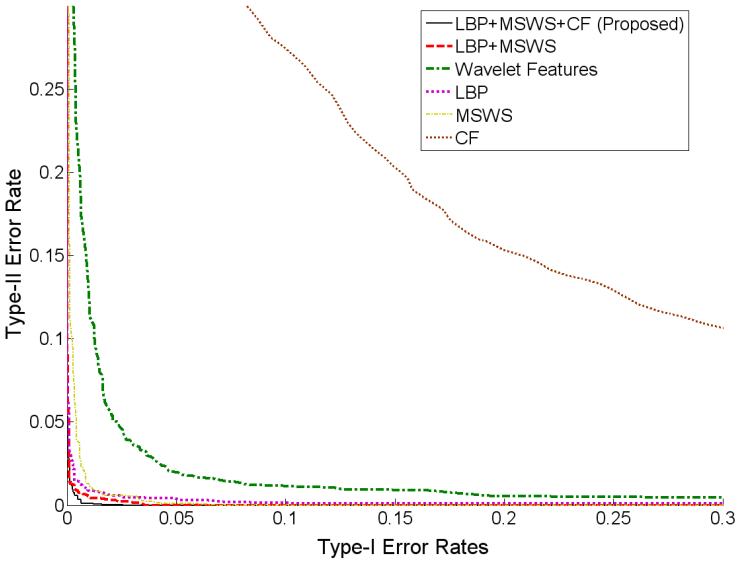


Fig. 5-7 Enlarged Receiver Operating Characteristics (ROC) Curves for Identification of Recaptured Images Using Different Feature Sets.

Table 5-2 Performance Comparison in Recaptured Image Identification in Terms of Equal Error Rate (EER), Where the Proposed Feature Set is Highlighted with Bold Face

Features	Dimension	EER (%)	EER Threshold
LBP	80	0.9	0.55
MSWS	54	1.1	0.35
CF	21	17.4	0.47
LBP+MSWS	134	0.7	0.43
LBP+MSWS+CF	**155**	**0.5**	**0.50**
Wavelet Stats [62, 64]	216	3.4	0.43

levels. Compared with image-details curve of the original image, the recapturing process typically results in flatter image-details characteristic curves.

Color Features (CF): Though majority of color artifacts can be eliminated with the best controllable settings, the color of our carefully recaptured images still looks different from their original images. Typically, the color of a recaptured image appears a bit washed out and more easily saturated. To capture these color anomalies, we compose a set of 21 color features including 3 average pixel values, 3 pairs correlations, 3 neighbor distribution centers of mass, 3 pairs energy ratios [26] from the RGB color space and 9 color moments computed from HSV color space [65].

5.4.2 Identification Experiment Using Reliable Image Sources

To test the effectiveness of the proposed features, we have set up an image dataset containing 2000 natural photos and 2700 carefully recaptured photos described in Section 5.2. Out of the natural photos, 300 photos are taken with the same 3 cameras used in image recapturing in Section 5.2 and the remaining 1700 photos are taken with 9 other cameras from 5 different brands including Canon, Casio, Lumix, Nikon and Sony. For both classes, 80% of the photos are randomly selected for training to ensure the classifier is well-trained and the remaining for testing. Such random apportion is repeated for 5 times so that we have 5 different combinations of training and testing datasets. The various types of features are then computed from the cropped central block of 1024×1024 from each photo.

After feature extraction, we train a probabilistic support vector machine (PSVM) classifier using the LIBSVM tools [147] by following the guild in [146]. By averaging the test results from the five different apportions, the various feature sets are compared in Fig. 5-7 in terms of the ROC curves and in Table 5-2 in terms of the equal error rates (EERs). Based on the results, we can see both the LBP and the MSWS features perform very well by giving low EERs of about 1.0%. This suggests both types of features are highly effective in capturing the artifacts introduced by the image recapturing process. By combining the LBP, MSWS and CF features, the identification performance can be improved further to a very low EER of 0.5%. We have also compared our proposed features with the wavelet statistics features in [62, 64]. It should be noted that according to [63], these wavelet statistics features have better performance than the geometry features [63] in identification of recaptured CG images on LCD screens. Hence we consider these wavelet statistics are the best statistical features in the literature for identification of the recaptured scenery. The comparison result shows that our combined features have clear advantage over these wavelet statistics features, where the EER of our combined features is 5.8 times lower than that of the wavelet statistics features.

With our trained PSVM classifier based on the combined features, we also classify the 50 survey photos used in human survey in Section 5.3. All the 50 photos are correctly classified.

Table 5-3 Performance Comparison When *Flickr* Photos are Used to Represent the Natural Category, Where the Proposed Feature Set is Highlighted with Bold Face

Features	Dimension	EER (%)
LBP	80	1.59
MSWS	54	1.54
CF	21	20.3
LBP+MSWS+CF	**155**	**1.21**
Wavelet Stats [62, 64]	216	3.38

5.4.3 Identification Experiment Using Internet Image Sources

In this experiment, we replace the 2000 natural photos from reliable camera sources in the previous experiment in Section 5.4.2 with 1000 Internet downloaded photos from *Flickr* website. These *Flickr* photos sources span 18 major camera brands and over 75 different camera models [130]. Similar to the previous identification experiment, we make 5 random apportions with 80% training data and 20% test data for each apportion. Through PSVM training and classification, comparison of the obtained EER rates based on different feature sets is tabulated in Table 5-3. From the results, the EER of 1.21% for our proposed feature combination shows better performance than the EER of 3.38% for the conventional wavelet statistics features.

Based on one selected apportion, we have also tested different subsets and combinations of the LBP and MSWS features for identification of the recaptured images. Table 5.4 shows the EERs for different feature sets. For LBP features, we find that the best result is achieved for the combined set when the four LBP operators, $LBP_{8,1}^{riu2}$, $LBP_{16,2}^{riu2}$, $LBP_{24,3}^{riu2}$ and $LBP_{24,4}^{riu2}$, are all included. For the MSWS features, the best performance is achieved when the combined feature set includes three decomposition levels $n \in \{1,2,3\}$ and all three color channels $c \in \{\text{red}, \text{green}, \text{blue}\}$. This result supports our current selection of the LBP and MSWS features.

5.5 Summary

Table 5-4 Performance Comparison for Different LBP and MSWS Feature Subsets and Combinations

Features	Remarks	Dimension	EER (%)
LBP	$LBP_{8,1}^{riu2}$	10	6.38
	$LBP_{16,2}^{riu2}$	18	4.50
	$LBP_{24,3}^{riu2}$	26	5.50
	$LBP_{24,4}^{riu2}$	26	4.50
	$LBP_{8,1}^{riu2} + LBP_{16,2}^{riu2} + LBP_{24,3}^{riu2}$	54	0.74
	$LBP_{16,2}^{riu2} + LBP_{24,3}^{riu2} + LBP_{24,4}^{riu2}$	70	2.00
	$LBP_{8,1}^{riu2} + LBP_{16,2}^{riu2} + LBP_{24,3}^{riu2} + LBP_{24,4}^{riu2}$	**80**	**0.5**
MSWS	$n \in \{1,2,3,4\}$ and $c \in \{\text{red}\}$	24	1.85
	$n \in \{1,2,3,4\}$ and $c \in \{\text{green}\}$	24	1.85
	$n \in \{1,2,3,4\}$ and $c \in \{\text{blue}\}$	24	0.74
	$n \in \{1,2,3\}$ and $c \in \{\text{red, blue, green}\}$	**54**	**0.54**
	$n \in \{2,3,4\}$ and $c \in \{\text{red, blue, green}\}$	54	2.96
	$n \in \{1,2,3\}$ and $c \in \{\text{blue, green}\}$	36	0.74
	$n \in \{1,2,3,4\}$ and $c \in \{\text{red, blue, green}\}$	96	0.56

In this chapter, we study identification of the carefully recaptured images on LCD screens. Through proper setup of the image recapturing environment and by tuning the controllable settings, we can recapture the artificial sceneries displayed on LCD screens with reasonably good quality. In a human identification study, we notice that such recaptured images can hardly be identified successfully by human eyes. Hence, these finely recaptured photos potentially post a threat for image forgers to both walk around the current forgery detection system and to fool human eyes. To prevent such a security loophole, we propose using several types of statistical features for classification of the recaptured images from natural images. Our proposed features capture the textured patterns, the loss-of-fine-details characteristics and the color anomalies introduced in the image recapturing process. Comparison results show that the proposed features works extremely well and they outperform the state-of-arts forensics features in identifying the finely recaptured images on LCD screens.

Chapter 6 Conclusions and Future Works

6.1 Conclusions

The digital revolution in the recent decades has brought us numerous unprecedented hardware and software tools, which make photo forgery an easy task. Consequently, a growing number of fraudulent cases involving image forgery have appeared in Internet and in various mass media during the recent years. The negative impact caused by the numerous photographic deceptions has gradually deteriorated the traditional trustworthiness on the digital photos which makes our "seeing no longer believing". As photos are still widely served as important visual evidences of real happenings and records of history in many important occasions, forensics analyses to tell the origin, integrity and authenticity of a photo are urgently needed.

In this thesis, we study several image forensics problems through statistical detection of some intrinsic image regularities and tampering anomalies. We investigate and propose algorithms that fall into the passive category, where prior information hiding is not needed. Specifically, the forensics problems we address are:

Accurate Detection of Demosaicing Regularity

In Chapter 2, we propose a novel detection framework of image demosaicing regularity. By addressing the common differences between demosaicing algorithms, our

method accurately estimates the underlying demosaicing formulas from the output reconstructed color images. Through a two-level reverse classification, particularly an expectation maximization reverse classification (EMRC) algorithm for resolving ambiguous demosaicing axes, we divide all demosaiced color samples into 16 categories, where samples reconstructed with the same or very similar demosaicing formulas are placed into the same category. This reduces the estimation errors of the demosaicing formulas due to classification errors. Based on partial second-order derivative correlation models, we propose to estimate the demosaicing weights for each category by considering both the intra-color channel and the cross-channel correlations using a regularized least square solution. Compared with the conventional pixel correlation models, which are commonly employed in the existing demosaicing detection methods [27-29, 45, 75], our correlation model based on the partial image derivatives can estimate the demosaicing weights better with less scenery-dependant interferences. This is because the derivatives used in our correlation models do not include the local DC components, which are largely scenery dependant. Comparison results show that our estimated demosaicing formulas can regenerate the demosaiced samples from the sensor samples with better accuracy than the intra-channel method in [45] for diversified demosaicing algorithms. To represent the image demosaicing regularity as general forensics features, we further propose to compute several demosaicing features sets including the weights, the error cumulants and the normalized group sizes. With a probabilistic support vector machine (PSVM) classifier and radial basis function (RBF) kernel, the comparison results show that our demosaicing features used outperform two previous detection methods [45, 75] in identifying 16 diversified demosaicing algorithms in the presence of common post-demosaicing processing. Moreover, we have also demonstrated that our demosaicing features are more sensitive to small image content variations than the intra-channel method [45] and can be efficiently used to classify different post-demosaicing processes with a fixed Hamilton's demosaicing algorithm.

Identification of Various Image Source Models

On top of our proposed demosaicing detection framework, we have applied our detected demosaicing features into the identification of various common image sources.

In Chapter 2, with sequential forward floating search (SFFS) feature selection, our reduced demosaicing features achieve extremely good performances in identifying digital still cameras (DSC) models and RAW-tools. With 250 selected features, we achieve an accuracy of 97.5% in identifying 14 digital still cameras of different models. With 50 selected features, we achieve an accuracy of 99.1% in identifying 10 commercial RAW tools. A large percentage of about 80% of the SFFS selected features are the weights features, followed by error cumulants and normalized group sizes. The selected features are typically distributed evenly among the different demosaicing categories. In Chapter 3, we identify the low-end mobile cameras from their output images. We propose to apply an eigenfeature regularization and extraction to identify a highly-discriminant subspace to transform our high-dimensional demosaicing features into a compact set of eigen features. For the identification of 9 mobile cameras of dissimilar models, we find that our identification error rates stabilize to a low level of about 1% only after only 8 eigen demosaicing features are included. Based on the same number of extracted eigen features, comparison results show that our demosaicing features outperform several conventional forensics features and the suggested combinations [49, 57, and 62]. By including cameras of the same model or very similar models, we find in the confusion matrix of 15-cam identification that, to some extent, our identification accuracies tend to mix among the cameras of the same model or very similar models. These good identification performances for various practical image sources are achieved based on large number of images blocks cropped at several fixed photo locations. Hence our image datasets contain a mixture of good and bad quality images. For the above identification of 14 DSCs, 10 RAW tools and 15 mobile cameras, the total number of cropped blocks used is respectively 33600, 24000 and 6000.

Universal Image Tampering Detection

By formulating the universal tampering as a large asymmetric binary classification problem in Chapter 4, we propose a universal tampering detection framework to simultaneously detect a wide range of common tampering types. From a syntactic experiment, we find that the common image tampering often modifies the pixel correlation where the distortion signature can be reliably captured into our statistically detected demosaicing features. By dividing our demosaicing features into a number of

small feature subsets, we learn a set of lightweight tampering detectors using PSVM with RBF kernel. These individual probabilistic classifiers exhibit diversified classification performances. To operate the classifiers at a common natural threshold of 0.5, we propose a nonlinear normalization using an exponential function. A novel FusionBoost learning is then proposed to construct a strong ensemble tampering detector from the normalized individual tampering detectors. Developed on top of the conventional RealBoost [121], FusionBoost adds a periodical backtracking mechanism called conditional toggling to repetitively switch off some unfavorable stages. Compared with another boosting algorithm with conditional exclusion in its backtracking, i.e. the FloatBoost [122], the conditional toggling of FusionBoost has several main differences: 1) Conditional toggling is made independent from the forward inclusion; 2) The *off*-ed stages are still kept which can be switched on again in the future. Such design differences avoid a deadlock problem faced by FloatBoost in combining a pre-trained set of probabilistic tampering detectors in order to improve the fusion performances. We also point out that the most frequent backtracking (period=1) does not necessarily lead to better performance. Instead, a larger period of 16, which is empirically determined, usually gives better fusion performance. Large-scale experiments show that our FusionBoost learned ensemble detector achieves very low average test error rates ranging from 2.0%-4.3% in detecting various forms of tampering for four diversified image sources. The ensemble detector works particularly well to detect various image filtering, resampling, additive noise, lossy JPEG compression and one-pixel shift. Though luminance domain operations and photomontage are relatively more difficult to detect, our results of less than 8% are still highly satisfactory. Comparison results also show that our proposed FusionBoost is highly effective in constructing a classifier ensemble. Consistent better performances in combining 20 individual probabilistic classifiers are achieved by our proposed FusionBoost than by other ensemble learning or several conventional classifier fusion strategies. Moreover, the classifier selection in FusionBoost eliminates the ineffective or redundant individual tampering detectors and the corresponding feature subsets, which reduces the system complexity.

Prevention of Image Recapturing Threat

Image recapturing often results in a poor image visual quality though it is known to potentially defeat many existing image forensics systems. None of the early works have studied whether image recapturing can result in images of reasonable visual quality and whether human beings can identify the recaptured images from the natural images simply based on the visual quality loss during the recapturing process. In Chapter 5, we study identification of the carefully recaptured images on LCD screens using commercial DSCs. Through proper setup of the image recapturing environment by tuning the controllable settings, we find the visual quality of recaptured images can be significantly improved over the casually recaptured images. In a human identification study, we find that our recaptured images can hardly be identified successfully by human eyes. Hence, these carefully recaptured images potentially post a threat for image forgers to both walk around the current passive forensics systems and to fool human eyes. To prevent such a security loophole, we propose several types of statistical features for identification of the recaptured images from the natural images. These features include local binary patterns (LBP), multi-scale wavelet statistics (MSWS) and color features (CF). Each type of features is designed to capture a specific anomaly, commonly accompanied with the image recapturing process. The LBP features capture the textures due to the tiny regular structures of LCD display. The loss-of-the-details is characterized by the MSWS features and the color anomalies are captured by the color features. With PSVM classification, experimental results show that the combination of our proposed features achieve low equal error rates of 0.5% for reliable photo sources and 1.21% for Internet photo sources. Through comparison, we show that our features consistently outperform the existing wavelet statistics features in identifying our carefully recaptured images on LCD screens.

6.2 Future Works

The technologies described in this thesis successfully address several important challenges in passive image forensics with extremely good results demonstrated based on large-scale tests. Since digital image forensics is still at its infancy with many

possible extensions, we briefly summarize some future directions below that may arise from our existing works.

1. In our demosaicing detection framework, the separation of sensor samples and demosaiced samples are based on Bayer color filter arrays (CFA). Though Bayer CFAs are dominantly used commercially and other CFAs, e.g. SuperCCD and complementary CFAs, share some similar properties as Bayer CFAs, the effectiveness of our demosaicing features shall be further evaluated for the cameras with non-Bayer CFAs. This will give a more comprehensive picture of using our proposed demosaicing features for general image forensics purposes;
2. Complementing our good results in identifying different image processing pipelines, additional features such as sensor noise patterns [34, 50] are needed for identification of individual camera devices. As highlighted in Chapter 2, our demosaicing features used do not perform as well for the uniform images but experience no performance degradation for heavily textured images. On the other hand, the sensor noise pattern based approaches work better in uniform images but not in heavily textured images [34]. We see good complementary merits in combining the two types of approaches for more reliable forensics analysis;
3. The current good source identification is still limited to 9-15 cameras. In mobile camera model identification, we have shown that the result can be adversely affected by increasing the number of cameras, especially those in the same or very similar models. Further investigation is needed to show how our forensics performance changes when the number of cameras increases substantially and to show the performance of identifying close camera models. Additional work can also be done to reliably build statistical template for a group of cameras which supposedly share very similar image processing pipelines, e.g. those cameras from the same model or the same brand. Such template shall be extendable and work well for the many unseen cameras of the same group;
4. In our proposed demosaicing detection, we have attempted to minimize the detection variations caused by different image content at the feature extraction level. However, dependence of the extracted statistical features and the image content can hardly be eliminated completely. We hypothesize that forensics decision for a test image can be better made based on some training images of similar image content

than based on images of very different content. Further investigation efforts to minimize the adverse forensics performance caused by different image content are needed at the forensics decision level;

5. Our current size of cropped image blocks is about 512×512. To make the forensics more challenging, the size of our cropped blocks shall be further reduced. For computing our demosaicing features on a reduced image area, generally we would still have enough demosaiced samples for statistically estimating the underlying demosaicing regularity. For example, a total of 32,768 demosaiced samples are available for a reduced block size of 128×128, which is about 26.3 times of our 1248 demosaicing weights to be estimated. However, we would expect a drop in our forensics accuracy as a larger percentage of uniform blocks would be generated with a smaller block size based on the same fixed-location cropping scheme that we employed in this thesis. We have shown that our demosaicing features do not perform as excellently on the uniform images as on the images with rich content variations. On another hand, the good forensics accuracy achieved with small block sizes will increase the sensitivity of our forensics analysis and to detect small tampering areas. This potential trade-off between our forensics accuracy and the sensitivity related with different block sizes shall be further investigated;
6. Attacks on passive image forensics techniques such as tamper hiding techniques [124-126] have also emerged in recent years. Evaluation of the security level against common attacks for our current forensics systems and designing attack-resistant forensics techniques are promising future directions;
7. Our current identification of recaptured images is still limited to using DSCs as the image acquisition devices and using the ubiquitous LCD as the display media. Other image recapturing threats using different recapturing devices, such as scanners and the low-end mobile cameras, and using different display media, such as high-quality printing and projection, shall be investigated in the future. Besides preventing the image recapturing threat, it is also interesting to identify the different recapturing pipelines and to estimate the parameters used in the recapturing.
8. Our current source model identification is based on less than twenty DSC models, RAW tools or mobile camera models. Since exhausting all different models and different types of image acquisition devices are difficult and may not be necessary,

one possible future addition would be constructing a category named “others”. The issues on how to construct a representative “others” category and on how to incorporate it into our classification-based source model identification framework shall be addressed in the future works.

Appendix A Compute Second-Order Derivative on 1D Periodical CFA Lattice based on Fourth- and Sixth-Order Approximations

As illustrated in Fig. 2-2, we let $\{F(n), n=1, 2,\ldots\}$ denote the discrete samples from a smooth continuous function $f(t)$ at equal sampling intervals. Based on the 1D periodical mosaic CFA lattice (Period =2), we can associate the indexes $\{\ldots, q\text{-}1, q+1, \ldots\}$ and $\{\ldots, q\text{-}2, q, q+2, \ldots\}$ respectively with the sensor samples and the demosaiced samples. Suppose $F(q)$ is demosaiced along t-axis, it is reasonable to assume $f(t)$ is smooth at $t=q$ such that both the left and the right derivatives of $f(t)$ are continuous at $t=q$, and Taylor series expansions are applicable below.

$$F(q+d) = F(q) + \frac{f^1(q)}{1!}d + \frac{f^2(q)}{2!}d^2 + \frac{f^3(q)}{3!}d^3 + \cdots \qquad \text{(A.1)}$$

In Section 2.3.1, we have derived the second-order derivative formula based on second-order approximation on the Taylor series expansions, where only the nearest sensor samples $F(q\text{-}1)$ and $F(q+1)$ are considered. In this appendix, we further derive the second-order derivative formulas based on fourth- and sixth-order approximations respectively by considering a larger neighborhood of sensor samples.

Fourth-Order Approximation

With fourth-order approximation on the Taylor series expansion, we consider a neighborhood of four sensor samples centered at $t=q$, including sensor samples $F(q\text{-}3)$, $F(q\text{-}1)$, $F(q+1)$ and $F(q+3)$ to have the following equations

$$\begin{cases} F(q+3) = F(q) + 3f^{(1)}(q) + 9f^{(2)}(q)/2 + 9f^{(3)}(q)/2 + 27f^{(4)}(q)/8 \\ F(q+1) = F(q) + f^{(1)}(q) + f^{(2)}(q)/2 + f^{(3)}(q)/6 + f^{(4)}(q)/24 \\ F(q-1) = F(q) - f^{(1)}(q) + f^{(2)}(q)/2 - f^{(3)}(q)/6 + f^{(4)}(q)/24 \\ F(q-3) = F(q) - 3f^{(1)}(q) + 9f^{(2)}(q)/2 - 9f^{(3)}(q)/2 + 27f^{(4)}(q)/8 \end{cases} \tag{A.2}$$

Through eliminating $f^{(1)}(q), f^{(3)}(q)$ and $f^{(4)}(q)$ in Eqn (A.2), we derive

$$f^{(2)}(q) = \frac{1}{72}\left(-F(q+3) + 81F(q+1) - 160F(q) + 81F(q-1) - F(q-3)\right) \tag{A.3}$$

Or equivalently,

$$F(q) = \boldsymbol{\alpha}^T \boldsymbol{\gamma} - \beta f^{(2)}(q) \tag{A.4}$$

where $\boldsymbol{\alpha} = \frac{1}{160}\begin{bmatrix} -1 & 81 & 81 & -1 \end{bmatrix}^T$, $\boldsymbol{\gamma} = \begin{bmatrix} F(q+3) & F(q+1) & F(q-1) & F(q-3) \end{bmatrix}$ and $\beta = 9/20$.

Sixth-Order Approximation

With sixth-order approximation on the Taylor series expansion, we consider a neighborhood of six sensor samples centered at $t=q$, including sensor samples $F(q\text{-}5)$, $F(q\text{-}3)$, $F(q\text{-}1)$, $F(q+1)$, $F(q+3)$ and $F(q+5)$ to write the following equations

$$\begin{cases} F(q+5)=F(q)+5f^{(1)}(q)+25f^{(2)}(q)/2+125f^{(3)}(q)/6+625f^{(4)}(q)/24 \\ +625f^{(5)}(q)/24+3125f^{(6)}(q)/144 \\ \\ F(q+3)=F(q)+3f^{(1)}(q)+9f^{(2)}(q)/2+9f^{(3)}(q)/2+27f^{(4)}(q)/8 \\ +81f^{(5)}(q)/40+81f^{(6)}(q)/80 \\ \\ F(q+1)=F(q)+f^{(1)}(q)+f^{(2)}(q)/2+f^{(3)}(q)/6+f^{(4)}(q)/24 \\ +f^{(5)}(q)/120+f^{(6)}(q)/720 \\ \\ F(q-1)=F(q)-f^{(1)}(q)+f^{(2)}(q)/2-f^{(3)}(q)/6+f^{(4)}(q)/24 \\ -f^{(5)}(q)/120+f^{(6)}(q)/720 \\ \\ F(q-3)=F(q)-3f^{(1)}(q)+9f^{(2)}(q)/2-9f^{(3)}(q)/2+27f^{(4)}(q)/8 \\ -81f^{(5)}(q)/40+81f^{(6)}(q)/80 \\ \\ F(q-5)=F(q)-5f^{(1)}(q)+25f^{(2)}(q)/2-125f^{(3)}(q)/6+625f^{(4)}(q)/24 \\ -625f^{(5)}(q)/24+3125f^{(6)}(q)/144 \end{cases} \tag{A.5}$$

Through eliminating $f^{(1)}(q), f^{(3)}(q), f^{(4)}(q), f^{(5)}(q)$ and $f^{(6)}(q)$, we derive

$$f^{(2)}(q)=\frac{1}{28800}\begin{pmatrix} 27F(q+5)-625F(q+3)+33750F(q+1)-66304F(q) \\ +33750F(q-1)-625F(q-3)+27F(q-5) \end{pmatrix} \tag{A.6}$$

Or equivalently,

$$F(q)=\boldsymbol{\alpha}^T\boldsymbol{\gamma}-\beta f^{(2)}(q) \tag{A.7}$$

where $\boldsymbol{\gamma}=\begin{bmatrix} F(q+5) & F(q+3) & F(q+1) & F(q-1) & F(q-3) & F(q-5) \end{bmatrix}$,

$\boldsymbol{\alpha}=\frac{1}{66304}\begin{bmatrix} 27 & -625 & 33750 & 33750 & -625 & 27 \end{bmatrix}^T$, and $\beta=225/518$.

Appendix B RealBoost

The idea of boosting is to use weak classifiers to construct a highly accurate prediction classifier by calling the weak classifiers repeatedly on different distributions over the training samples. RealBoost [121] is real-valued version of the discrete adaptive boost [120] or AdaBoost. For a two-class problem, let $(x_1, z_1), \ldots, (x_N, z_N)$ denotes a set of N labeled training images, where x_n is the n^{th} image and $z_n \in \{-1, +1\}$ is the class label associated with x_n. A stronger classifier is written as a linear combination of Q weak classifiers

$$C_Q(x) = \sum_{q=1}^{Q} \omega_q c_q(x) \tag{B.1}$$

where $c_q(x)$ is a weak classifier learned in the q^{th} iteration for predicting the class label z and ω_q is its associated classifier weight. The predicted label for x based on the strong classifier is obtained by $C(x) = \text{sign}\left[C_Q(x)\right]$ and $\left|C_Q(x)\right|$ indicates the confidence of classification decision. An error occurs when $C(x) \neq z$ or $zC_Q(x) < 0$.

The detailed RealBoost algorithm is described in Fig. B-1. In the initialization, each training image x_n is assigned a weight denoting its importance in training and finding the next weak classifier. The weights distribution is maintained and updated when each new classifier is added. The weight updating formula in Eqn (B.2) aims to place more

- Given labeled training $(x_1, z_1), ..., (x_n, z_n), ..., (x_N, z_N)$; x_n is the n^{th} image, $z_n \in \{-1, +1\}$ is the class label;
- Initialize $D_n^{(0)} = \frac{1}{N}$;
- For $q = 0, ..., Q$:
  - Train weak classifiers using distribution $\{D_n^{(q)}\}$;
  - Select and get weak classifier $c_q(x)$ for predicting the *z*, the label of *x*;
  - Choose ω_q, the weight assigned to classifier $c_q(x)$;
  - Update:

$$D_n^{(q+1)} = \frac{D_n^{(q)} \exp\left(-\omega_q z_n c_q(x_n)\right)}{Z_q} \tag{B.2}$$

    where

$$Z_q = \sum_{n=1}^{N} D_n^{(q)} \exp\left(-\omega_q z_n c_q(x_n)\right) \tag{B.3}$$

    is a normalization factor.
- Output the strong classifier:

$$C(x) = \text{sign}\left(C_Q(x)\right) = \text{sign}\left(\sum_{q=1}^{Q} \omega_q c_q(x)\right) \tag{B.4}$$

Fig. B-1 Two-Class RealBoost

emphasis on the training images which are classified erroneously or with a low confidence previously. The work [121] has proved that the classification error rate for the training images achieved by C_Q is upper bounded by a loss function [121]

$$J(C_Q) = \prod_{q=1}^{Q} Z_q \tag{B.5}$$

Note that $D_n^{(q+1)}$ in Eqn (B.2) can be rewritten as

$$\begin{aligned} D_n^{(q+1)} &= D_n^{(q)} \frac{\exp\left(-\omega_q z_n c_q\left(x_n\right)\right)}{Z_q} \\ &= D_n^{(q-1)} \frac{\exp\left(-\omega_{q-1} z_n c_{q-1}\left(x_n\right)\right)}{Z_{q-1}} \frac{\exp\left(-\omega_q z_n c_q\left(x_n\right)\right)}{Z_q} \\ &= \ldots \\ &= D_n^{(0)} \frac{\exp\left(-\sum_{p=0}^{q} \omega_p z_n c_p\left(x_n\right)\right)}{\prod_{p=0}^{q} Z_p} \\ &= D_n^{(0)} \frac{\exp\left(-z_n C_q\left(x_n\right)\right)}{\prod_{p=0}^{q} Z_p} \end{aligned} \tag{B.6}$$

The normalized weights shall satisfy

$$\sum_{n=1}^{N} D_n^{(q+1)} = 1 \tag{B.7}$$

By substituting Eqn (B.6) in to Eqn (B.7) and reorganizing the terms, we can derive

$$\prod_{p=0}^{q} Z_p = \sum_{n=1}^{N} D_n^{(0)} \exp\left(-z_n C_q\left(x_n\right)\right) \tag{B.8}$$

Therefore, the upper bound of average error rate in (B.5) is equivalently to be expressed as the exponential loss function

$$J\left(C_Q\right) = \prod_{q=1}^{Q} Z_q = \sum_{n=1}^{N} D_n^{(0)} \exp\left(-z_n C_Q\left(x_n\right)\right) \tag{B.9}$$

Note that in Eqn (B.9), the term $z_n C_Q\left(x_n\right)$ actually indicate the classification margin for the training image x_n. Suppose the current number of classifiers is $q-1$, the RealBoost constructs a new weak classifier $\omega_q c_q\left(x\right)$ by stagewise minimizing the loss function in Eqn (B.9). Generally, there are many ways to learn a weak classifier $c_q\left(x\right)$. One commonly adopted approach is to learn a large pool of weak classifiers based on the current distribution $\left\{D_n^{(q-1)}\right\}$, each based on a single feature or a feature block in the

feature space of *x*. The best weak learner $c_q(x)$ out of the classifier pool is then selected based on minimizing the loss function of $J(C_q)$. Given a new classifier $c_q(x)$, one important issue addressed by RealBoost is how to assign the weight ω_q associated with $c_q(x)$. The work [121] has suggested that ω_q can be found by minimizing Z_q in Eqn (B.3). Note that since $Z_0, Z_1, \ldots, Z_{q-1}$ are already fixed, minimizing Z_q is equivalent to minimize $J(C_q)$ in the current iteration. In order to minimize Z_q with respect to ω_q and suppose $c_q(x_n) \in \{-1, 0, +1\}$ has only three quantized output, Z_q in Eqn (B.3) can be re-expressed as

$$\begin{aligned} Z_q &= \sum_{n=1}^{N} D_n^{(q)} \exp\left(-\omega_q z_n c_q(x_n)\right) \\ &= W_0^{(q)} + W_-^{(q)} \exp\left(\omega_q\right) + W_+^{(q)} \exp\left(-\omega_q\right) \end{aligned} \tag{B.10}$$

where $W_0^{(q)} = \sum_{\forall n:\, z_n c_q(x_n)=0} D_n^{(q)}$, $W_-^{(q)} = \sum_{\forall n:\, z_n c_q(x_n)=-1} D_n^{(q)}$ and $W_+^{(q)} = \sum_{\forall n:\, z_n c_q(x_n)=+1} D_n^{(q)}$. By taking derivative of Z_q in Eqn (B.10) with respect to ω_q, we derive

$$\frac{dZ_q}{d\omega_q} = W_-^{(q)} \exp\left(\omega_q\right) - W_+^{(q)} \exp\left(-\omega_q\right) \tag{B.11}$$

At the minimum of Z_q, $\frac{dZ_q}{d\omega_q} = 0$ shall be satisfied. Therefore, the best ω_q is derived as

$$\omega_q = \frac{1}{2} \ln\left(\frac{W_+^{(q)}}{W_-^{(q)}}\right) \tag{B.12}$$

Since $W_+^{(q)}$ and $W_-^{(q)}$ can be very small values or even zero, as suggested in [121], it is advantageous to add a smoothing parameter ε in Eqn (B.12) in order to improve the numerical stability in computing ω_q. The new formula now becomes

$$\omega_q = \frac{1}{2}\ln\left(\frac{W_+^{(q)} + \varepsilon}{W_-^{(q)} + \varepsilon}\right) \tag{B.13}$$

By stagewise minimizing the exponential error function and improving the classification margins, the iterative learning procedures in RealBoost generally work well with good generalization performance to unseen data.

Appendix C RealBoost for Probabilistic Classifiers

As discussed in the two-class RealBoost in Appendix A, the training images $(x_1, z_1),...,(x_N, z_N)$ has two labels that $z_n = -1 \text{ or } +1$. The weak classifiers $\{c_q(x), q = 1,\ldots,Q\}$ are learned to predict the class label $z \in \{-1,+1\}$ of x. The term $\text{sign}(z_n c_q(x_n))$ is used to determine whether a correct classification is made, where a "+" sign indicates a correct classification and a "-" sign indicates an error. The classification margin can be written as $z_n c_q(x_n)$. The entire development of RealBoost is based on the above formulation. In this appendix, we extend the RealBoost for combining probabilistic classifiers $\{s_q(x), q = 1,\ldots,Q\}$, where $s_q(x) \in [0,1]$ is satisfied, and we re-derive several important RealBoost formulas for the probabilistic classifiers.

We first express the RealBoost-combined probabilistic classifier with Q individual classifiers as

$$S_Q(x) = \frac{\sum_{q=1}^{Q} \omega_q s_q(x)}{\sum_{q=1}^{Q} \omega_q} \tag{C.1}$$

where we can view that the q^{th} probabilistic classifier $s_q(x)$ as predicting the label $y \in \{0,1\}$ of an input image x instead of the label $z \in \{-1,+1\}$. Therefore, the following conversion formula holds

$$\begin{aligned} z &= 2y - 1 \\ c_q(x) &= 2s_q(x) - 1 \end{aligned} \tag{C.2}$$

From Eqn (C.2), it is easy to show that $s_q(x) = \frac{1}{2}\left(c_q(x) + 1\right)$. Substituting this into Eqn (C.1), we derive that

$$S_Q(x) = \frac{\sum_{q=1}^{Q} \omega_q s_q(x)}{\sum_{q=1}^{Q} \omega_q} = \frac{1}{2}\left(\frac{\sum_{q=1}^{Q} \omega_q c_q(x)}{\sum_{q=1}^{Q} \omega_q} + 1 \right) \tag{C.3}$$

Eqn (C.3) can be related with the combined RealBoost classifier in Eqn (B.1) and we substitute Eqn (B.1) into Eqn (C.3) to get

$$S_Q(x) = \frac{1}{2}\left(\frac{C_Q(x)}{\sum_{q=1}^{Q} \omega_q} + 1 \right) \tag{C.4}$$

Note that $C_Q(x)$ has same performance as its normalized version $\frac{C_Q(x)}{\sum_{q=1}^{Q} \omega_q}$ because

$$\text{sign}\left(zC_Q(x)\right) \equiv \text{sign}\left(z \frac{C_Q(x)}{\sum_{q=1}^{Q} \omega_q} \right) \tag{C.5}$$

Therefore, using the RealBoost procedures to improve the performance of $C_Q(x)$ would have the same impact on the error rate of the normalized ensemble probabilistic classifier $S_Q(x)$. Below, several RealBoost formulas are re-derived to cater for the probabilistic classifiers $s_1(x), \ldots, s_q(x), \ldots, s_Q(x)$.

Based on Eqn (C.1), the classification margin used in RealBoost can be re-written as

$$zc_q(x) = (2y - 1)\left(2s_q(x) - 1\right) \tag{C.6}$$

If $y = 0,$ by capitalizing the property that the probabilistic output $s_q(x) \in [0,1]$, we can derive from Eqn (C.6) that

$$\begin{aligned} zc_q(x) &= 1 - 2s_q(x) \\ &= 1 + 2\left(y - s_q(x)\right) \\ &= 1 - 2\left|y - s_q(x)\right| \end{aligned} \tag{C.7}$$

If $y = 1,$ we write from Eqn (C.6) that

$$\begin{aligned} zc_q(x) &= 2s_q(x) - 1 \\ &= 1 - 2\left(y - s_q(x)\right) \\ &= 1 - 2\left|y - s_q(x)\right| \end{aligned} \tag{C.8}$$

From Eqn (C.3) and Eqn (C.4), regardless of the value of *y*, we can write

$$zc_q(x) = 1 - 2\left|y - s_q(x)\right| \tag{C.9}$$

Substituting Eqn (C.9) and Eqn (B.1) into Eqn (B.9), the exponential loss function to be minimized can be expressed as

$$\begin{aligned} J(C_Q) &= \sum_{n=1}^{N} D_n^{(0)} \exp\left(-z_n C_Q(x_n)\right) \\ &= \sum_{n=1}^{N} D_n^{(0)} \exp\left(-\sum_{q=1}^{Q} z_n c_q(x_n)\right) \\ &= \sum_{n=1}^{N} D_n^{(0)} \exp\left(-\sum_{q=1}^{Q}\left(1 - 2\left|y - s_q(x)\right|\right)\right) \end{aligned} \tag{C.10}$$

The weight update formula in Eqn (B.2) can also be re-expressed as

$$\begin{aligned} D_n^{(q+1)} &= \frac{D_n^{(q)} \exp\left(-\omega_q z_n c_q\left(x_n\right)\right)}{Z_q} \\ &= \frac{D_n^{(q)} \exp\left(-\omega_q \left(1-2\left|y-s_q\left(x_n\right)\right|\right)\right)}{} \\ &= D_n^{(q)} \exp\left(2\omega_q \left|y-s_q\left(x_n\right)\right|\right) \frac{\exp\left(-\omega_q\right)}{Z_q} \end{aligned} \tag{C.11}$$

The last term $\dfrac{\exp\left(-\omega_q\right)}{Z_q}$ of Eqn (C.11) can be treated as a single normalization constant.

The classifier weight ω_q can be written below or in Eqn (B.13)

$$\omega_q = \frac{1}{2}\ln\left(\frac{W_+^{(q)}+\varepsilon}{W_-^{(q)}+\varepsilon}\right) \tag{C.12}$$

However, the terms $W_+^{(q)}$ and $W_-^{(q)}$ are re-written below by substituting Eqn (C.9)

$$\begin{aligned} W_-^{(q)} &= \sum_{\forall n:\, z_n c_q(x_n)<0} D_n^{(q)} = \sum_{\forall n:\left|y_n - s_q(x_n)\right|>0.5} D_n^{(q)} \\ W_+^{(q)} &= \sum_{\forall n:\, z_n c_q(x_n)>0} D_n^{(q)} = \sum_{\forall n:\left|y_n - s_q(x_n)\right|<0.5} D_n^{(q)} \end{aligned} \tag{C.13}$$

Author's Publication

Journal Papers

H. Cao, and A.C. Kot, "Accurate Detection of Demosaicing Regularity for Digital Image Forensics," *IEEE Trans. on Information Forensics and Security*, vol. 4(4), pp. 899-910, Dec. 2009

H. Cao, and A.C. Kot, "Lossless Data Embedding to Secure Electronic Inks," *IEEE Trans. on Information Forensics and Security*, vol. 5(2), pp. 314-323, June 2010

H. Cao, and A.C. Kot, "FusionBoosted Ensemble Image Tampering Detection," *submitted to IEEE Trans. on Information Forensics and Security,* 2010

H. Cao, and A.C. Kot, "Detection of Tampering Inconsistencies on Mobile Photos," *accepted in IWDW Workshop (Best Paper Award) and will appear in LNCS Journal,* 2010

H. Cao, and A.C. Kot, "On Establishing Edge Adaptive Grid for Bi-Level Image Data Hiding," *to be submitted to an IEEE Journal,* 2010

Conference Papers

H. Cao, and A.C. Kot, "Identification of Recaptured Photographs on LCD Screens," *in Proc. of ICASSP*, pp. 1790-1793, 2010

H. Cao, and A.C. Kot, "Mobile Camera Identification Using Demosaicing Features," *in Proc. of ISCAS*, pp. 1683-1686, 2010

H. Cao, and A.C. Kot, "Accurate Detection of Demosaicing Regularity from Output Images," in *Proc. of ISCAS*, pp. 497-500, 2009

H. Cao, and A.C. Kot, "Lossless Data Hiding for Electronic Ink," in *Proc. of ICASSP*, pp. 1381-1384, 2009

H. Cao, and A.C. Kot, "RAW-Tool Identification through Detected Demosaicing Regularity," in *Proc. of ICIP*, pp. 2885-2888, 2009

H. Cao, and A.C. Kot, "A Generalized Model for Detection of Demosaicing Characteristics," *in Proc. of ICME*, pp. 1513-1516, 2008

H. Cao, and A.C. Kot, "Online Structure Based Chinese Character Pre-Classification," in *Proc. of ICPR*, vol. 2, pp. 395-398, 2004

H. Cao, and A.C. Kot, "Modified Kohonen Learning Network and its Application in Chinese Character Recognition," *in Proc. of IEEE TENCON*, vol. 2, pp. 136-139, 2004

J. Cheng, A.C. Kot, J. Liu and H. Cao, "Steganalysis on Binary Text Images," *in Proc. of ICASSP*, vol. 4, pp. 689-692, 2005

J. Cheng, A.C. Kot, J. Liu and H. Cao, "Steganalysis of Data Hiding in Binary Text Images," *in Proc. of ISCAS*, vol. 5, pp. 4405-4408, 2005

J. Cheng, A.C. Kot, J. Liu and H. Cao, "Detection of Data Hiding in Binary Text Images," *in Proc. of ICIP*, vol. 3, pp. 73-76, 2005

Bibliography

[1] D.A. Brugioni, *Photo Fakery: the History and Techniques of Photographic Deception and Manipulation*, Dulles, Va.: Brassey's, 1999

[2] "Spanish MP's Photo Used for Osama Bin Laden Poster," in *BBC News*, 16 Jan, 2009

[3] "Israel: Women Photoshopped from Cabinet Picture to Cater to the Ultra-Orthodox," in *The Huffington Post*, 18 Jul, 2009

[4] B.V.D. Beek, "Photos Cheats," *in The Singapore Straits Times*, 2006

[5] D. Cyranoski, "Verdict: Hwang's Human Stem Cells were all Fakes," *Nature*, vol. 439, pp. 122-123, 2006

[6] S. Graham, "10 Famous Fake Photos," *in Bright Hub-Digtal Photography*, 2009

[7] H. Farid, "Photo Tampering Throughout the History," Available: http://www.cs.dartmouth.edu/farid/research/digitaltampering/

[8] F. Mosleh (Kodak), "Cameras in Handsets Evolving from Novelty to DSC Performance, Despite Constraints," Embedded.com, 2008

[9] T. Douglas, "Shaping the Media with Mobiles," *in BBC News*, 2005

[10] J. Lewis, "Don't Just Stand and Stare, Shoot it, Too", *in The Singapore Straits Times*, 28 April, 2007

[11] H. Pearson, "Forensic Software Traces Tweaks to Images," Nature, vol. 439, pp. 520-521, 2006

[12] T. Beier, and S. Neely, "Feature-Based Image Metamorphosis," *in Proc. of ACM SIGGRAPH Computer Graphics*, vol. 26(2), pp. 35-42, 1992

[13] J. Adams, K. Parulski, and K. Spaulding, "Color Processing in Digital Cameras," *IEEE Micro*, vol. 18(6), pp. 20-30, 1998

[14] R. Ramanath, W.E. Snyder, Y. Yoo, and M. S. Drew, "Color Image Processing Pipeline," *IEEE Signal Processing Magazine*, vol. 22(1), pp. 34-43, Jan 2005

[15] G.L. Friedman, "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image," *IEEE Trans. on Consumer Electronics*, vol. 39, pp. 905-910, 1993

[16] P. Blythe, and J. Fridrich, "Secure Digital Camera," *in Proc. of Digital Forensic Research Workshop (DFRWS)*, 2004

[17] J. Cox, M.L. Miller, and J.A. Bloom, *Digital Watermarking*, CA: Morgan Kaufmann, 2001

[18] H. Cao and A.C. Kot, "Lossless Data Hiding for Electronic Inks," *in Proc. of ICASSP*, pp. 1381-1384, 2009

[19] F. Xiao, J.E. Farrell, J. DiCarlo, and B. Wandell, "Preferred Color Spaces for White Balancing," *in Proc. of SPIE*. vol. 5017, pp. 342-350, 2003

[20] G.K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Trans. on Consumer Electronics*, vol. 38(1), pp. xviii-xxxiv, 1992

[21] "JPEG Chroma Subsampling", Available: http://www.impulseadventure.com

[22] Z.J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for Identification of Images Acquired with Digital Cameras " *in Proc. of SPIE*. vol. 4232, p. 505, 2001

[23] K, Kurosawa, K. Kuroki, and N. Saitoh, "An Approach to Individual Video Camera Identification," *Journal of Forensic Sciences*, vol. 47(1), pp. 97-102, 2002

[24] N. Saitoh, K. Kurosawa, K. Kuroki, N. Akiba, Z.J. Geradts, and J. Bijhold, "CCD Fingerprint Method for Digital Still Cameras," *in Proc. of SPIE*. vol. 4709, pp. 37-48, 2002

[25] K. Kurosawa and N. Saitoh, "Fundamental Study on Identification of CMOS Cameras," *in Proc. of SPIE*. vol. 5108, p. 202, 2003

[26] M. Kharrazi, H.T. Sencar, and N. Memon, "Blind Source Camera Identification," *in Proc. of ICIP*. vol. 1, pp. 709-712, 2004

[27] S. Bayram, H.T. Sencar, N. Memon, and I. Avcibas, "Source Camera Identification Based on CFA Interpolation," *in Proc. of ICIP*. vol. 3, pp. 69-72, 2005

[28] S. Bayram, H.T. Sencar, and N. Memon, "Improvements on Source Camera-Model Identification Based on CFA Interpolation," *in WG 11.9 Int. Conf. on Digital Forensics*, 2006

[29] S. Choi, E.Y. Lam, and K.K.Y. Wong, "Source Camera Identification using Footprints from Lens Aberration," *in Proc. of SPIE*, p. 60690J, 2006

[30] S. Choi, E.Y. Lam, and K.K.Y. Wong, "Feature Selection in Source Camera Identification," *in Proc. of SMC*, vol. 4, pp. 3176-3180, 2006

[31] S. Choi, E.Y. Lam, and K.K.Y. Wong, "Source Camera Identification using Footprints from Lens Aberration," *in Proc. of SPIE*. vol. 6069, p. 60690J, 2006

[32] S. Choi, E.Y. Lam, and K.K.Y. Wong, "Automatic Source Camera Identification using the Intrinsic Lens Radial Distortion," *Optics Express*, vol. 14(24), pp. 11551-11565, 2006

[33] Y. Long and Y. Huang, "Image Based Source Camera Identification Using Demosaicing," *in Proc. of IEEE 8th Workshop on MSP*, pp. 419-424, 2006

[34] J. Lucas, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," *IEEE Trans. on Information Forensics and Security*, vol. 1(2), pp. 205-214, 2006

[35] M.-J. Tsai and G.-H. Wu, "Using Image Features to Identify Camera Sources," *in Proc. of ICASSP*, vol. 2, pp. 297-300, 2006

[36] Celiktutan, I. Avcibas, and B. Sankur, "Blind Identification of Cellular Phone Cameras," *in Proc. of SPIE*, vol. 6505, p. 65051H, 2007

[37] M. Chen, J. Fridrich, and M. Goljan, "Digital Imaging Sensor Identification (Further Study)," *in Proc. of SPIE*, vol. 6505, p. 65050P, 2007

[38] S.-H. Chen and C.-T. Hsu, "Source Camera Identification Based on Camera Gain Histogram," *in Proc. of ICIP*, vol. 4, pp. 429-432, 2007

[39] E. Dirik, H.T. Sencar, and N. Memon, "Source Camera Identification Based on Sensor Dust Characteristics," *in Proc. of IEEE SAFE*, pp. 1-6, 2007

[40] M. Goljan, M. Chen, and J. Fridrich, "Identifying Common Source Digital Camera from Image Pairs," *in Proc of ICIP*, pp. 125-128, 2007

[41] H. Gou, A. Swaminathan, and M. Wu, "Robust Scanner Identification Based on Noise Features," *in Proc. of SPIE*. vol. 6505, p. 65050S, 2007

[42] N. Khanna and A.K. Mikkilineni, "Scanner Identification Using Sensor Pattern Noise," *in Proc. of SPIE*. vol. 6505, p. 65051K, 2007

[43] N. Khanna, A.K. Mikkilineni, G.T.C. Chiu, J. P. Allebach, and E. J. Delp, "Forensic Classification of Imaging Sensor Types," *in Proc. of SPIE*, vol. 6505, p. 65050U, 2007

[44] Y. Sutcu, S. Bayram, H.T. Sencar, and N. Memon, "Improvements on Sensor Noise Based Source Camera Identification," *in Proc. of ICME*, pp. 24-27, 2007

[45] A. Swaminathan, M. Wu, and K.J.R. Liu, "Nonintrusive Component Forensics of Visual Sensors Using Output Images," *IEEE Trans. on Information Forensics and Security*, vol. 2(1), pp. 91-106, 2007

[46] M.-J. Tsai, C.-L. Lai, and J. Liu, "Camera/Mobile Phone Source Identification for Digital Forensics," *in Proc. of ICASSP*. vol. 2, pp. 221-224, 2007

[47] L.T. Van, S. Emmanuel, and M.S. Kankanhalli, "Identifying Source Cell Phone using Chromatic Aberration," *in Proc. of ICME*, pp. 883-886, 2007

[48] S. Bayram, H.T. Sencar, and N. Memon, "Classification of Digital Camera-Models based on Demosaicing Artifacts," *Digital Investigation*, vol. 5(1-2), pp. 49-59, 2008

[49] Celiktutan, B. Sankur, and I. Avcibas, "Blind Identification of Source Cell-Phone Model," *IEEE Trans. on Information Forensics and Security*, vol. 3(3), pp. 553-566, 2008

[50] M. Chen, J. Fridrich, M. Goljan, and J. Lucas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Trans. on Information Forensics and Security*, vol. 3(1), pp. 74-89, 2008

[51] A.E. Dirik, H.T. Sencar, and N. Memon, "Digital Single Lens Reflex Camera Identification From Traces of Sensor Dust," *IEEE Trans. on Information Forensics and Security*, vol. 3, pp. 539-552, 2008

[52] E.J. Alles, Z.J.M.H. Geradts, and C.J. Veenman, "Source Camera Identification for Low Resolution Heavily Compressed Images," *in Proc. of ICCSA*, pp. 557-567, 2008

[53] T. Filler, J. Fridrich, and M. Goljan, "Using Sensor Pattern Noise for Camera Model Identification," *in Proc. of ICIP*, pp. 1296-1299, 2008

[54] C. Gallagher and T. Chen, "Image Authentication by Detecting Traces of Demosaicing," *in Proc. of CVPRW*, pp. 1-8, 2008

[55] M. Goljan and J. Fridrich, "Camera Identification from Cropped and Scaled Images," *in Proc. of SPIE*. vol. 6819, p. 68190E, 2008

[56] S. McCloskey, "Confidence Weighting for Sensor Fingerprinting," i*n Proc. of CVPRW apos*, pp. 1-6, 2008

[57] C. McKay, A. Swaminathan, H. Gou, and M. Wu, "Image Acquisition Forensics: Forensic Analysis to Identify Imaging Source," *in Proc. of ICASSP*, pp. 1657-1660, 2008

[58] C. Zhang and H. Zhang, "Digital Camera Identification Based on Canonical Correlation Analysis," *in Proc. of IEEE Workshop on MSP*, pp. 769-773, 2008

[59] A.E. Dirik, H.T. Sencar and N. Memon, "Flatbed Scanner Identification Based on Dust and Scratches over Scanner Platen," *in Proc. of ICASSP*, pp. 1385-1388, 2009

[60] C. Zhang and H. Zhang, "Digital Camera Identification Based on Curvelet Transform," *in Proc. of ICASSP*, pp. 1389-1392, 2009

[61] J. Bloy, "Blind Camera Fingerprinting and Image Clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30(3), pp. 532-534, 2008

[62] S. Lyu and H. Farid, "How Realistic is Photorealistic?," *IEEE Trans. on Signal Processing*, vol. 53(2), pp. 845-850, 2005

[63] 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 Proc. of ACM Int. Conf. on Multimedia*, pp. 239-248, 2005

[64] S. Lyu, "Natural Image Statistics for Digital Image Forensics," *ph.D thesis*, Dartmouth College, 2005

[65] Y. Chen, Z. Li, M. Li, and W.-Y. Ma, "Automatic Classification of Photographs and Graphics," *in Proc. of ICME*, vol. 9(12), pp. 973-976, 2006

[66] Y. Wang and P. Moulin, "On Discrimination between Photorealistic and Photographic Images," *in Proc. of ICASSP*, vol. 2, pp. 161-164, 2006

[67] J. Fridrich, D. Soukal, and J. Lucas, "Detection of Copy-Move Forgery in Digital Images," *in Proc. of DFRWS*, 2003.

[68] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A Classifier Design for Detecting Image Manipulations," *in Proc. of ICIP*. vol. 4, pp. 2645-2648, 2004.

[69] T.-T. Ng, and S.-F. Chang, "A Model for Image Splicing," *in Proc. of ICIP*, vol. 2, pp. 1169-1172, 2004.

[70] T.-T. Ng, S.-F. Chang, and Q. Sun, "Blind Detection of Photomontage Using High Order Statistics," *in Proc. of ISCAS*. vol. 5, pp. 688-691, 2004.

[71] A.C. Popescu and H. Farid, "Statistical Tools for Digital Forensics," *in Proc. of 6th Int. Workshop on Information Hiding*, 2004.

[72] A.C. Popescu, “Statistical Tools for Digital Image Forensics,” *ph.D Thesis*, Dartmouth College, 2004

[73] D.-Y. Hsiao and S.-C. Pei, "Detecting Digital Tampering by Blur Estimation," *in Proc. of First Int. Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 254-278, 2005

[74] M.K. Johnson and H. Farid, "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting," *in Proc. of ACM Multimedia Security Workshop*, pp. 1-10, 2005

[75] C. Popescu and H. Farid, "Exposing Digital Forgeries in Color Filter Array Interpolated Images," *IEEE Trans. on Signal Processing*, vol. 53, pp. 3948-3959, 2005

[76] C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Resampling," *IEEE Trans. on Signal Processing*, vol. 53, pp. 758-767, 2005

[77] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image Manipulation Detection," *Journal of Electronic Imaging*, vol. 15, p. 041102, 2006

[78] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting Doctored JPEG Images via DCT Coefficient Analysis," *in Lecture Notes in Computer Science*. vol. 3953: Springer Berlin / Heidelberg, pp. 423-435, 2006

[79] M.K. Johnson and H. Farid, "Exposing Digital Forgeries through Chromatic Aberration," in Proc. of ACM Multimedia Security Workshop, 2006

[80] J. Lucas, J. Fridrich, and M. Goljan, "Detecting Digital Image Forgeries Using Sensor Pattern Noise," *in Proc. of SPIE*, p. 60720Y, vol. 6072, 2006

[81] W. Luo, J. Huang, and G. Qiu, "Robust Detection of Region-Duplication Forgery in Digital Image," *in Proc. of ICPR*. pp. 746-749, vol. 4, 2006

[82] A. Swaminathan, M. Wu, and K.J.R. Liu, "Image Tampering Identification using Blind Deconvolution," *in Proc. of ICIP*, pp. 2309-2312, 2006

[83] D. Fu, Y.Q. Shi, and W. Su, "A Generalized Benford's Law for JPEG Coefficients and its Applications in Image Forensics," *in Proc. of SPIE*, vol. 6505, p. 65051L, 2007

[84] Y.-F. Hsu and S.-F. Chang, "Image Splicing Detection using Camera Response Function Consistency and Automatic Segmentation," *in Proc. of ICME*, pp. 28-31, 2007

[85] M.K. Johnson, "Lighting and Optical Tools for Image Forensics," *ph.D thesis*, Dartmouth College, 2007

[86] M.K. Johnson and H. Farid, "Exposing Digital Forgeries through Specular Highlights on the Eye," *in Proc. of Information Hiding*, pp. 311-325, 2007

[87] M.K. Johnson and H. Farid, "Exposing Digital Forgeries in Complex Lighting Environments," *IEEE Trans. on Information Forensics and Security*, vol. 2(3-1), pp. 450-461, 2007

[88] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A Novel Method for Detecting Cropped and Recompressed Image Block," *in Proc. of ICASSP*. vol. 2, pp. 217-220, 2007

[89] Y. Sutcu, B. Coskun, H.T. Sencar, and N. Memon, "Tamper Detection Based on Regularity of Wavelet Transform Coefficients," *in Proc. of ICIP*, pp. 397-400, 2007

[90] A. Swaminathan, M. Wu, and K.J.R. Liu, "Digital Image Forensics via Intrinsic Fingerprints," *IEEE Trans. on Information Forensics and Security*, vol. 3(1), pp. 101-117, 2008

[91] H. Farid, "Exposing Digital Forgeries from JPEG Ghosts," *IEEE Trans. on Information Forensics and Security*, vol. 4(1), pp. 154-160, 2009

[92] W.-H. Chuang, A. Swaminathan and M. Wu, "Tampering Identification Using Empirical Frequency Response," *in Proc. of ICASSP*, pp. 1517-1520, 2009

[93] S. Bayram, H.T. Sencar and N. Memon, "An Efficient and Robust Method for Detecting Copy-Move Forgery," *in Proc. of ICASSP*, pp. 1053-1056, 2009

[94] A.E. Dirik and N. Memon, "Image Tamper Detection Based on Demosaicing Artifacts," *in Proc. of ICIP*, pp. 1497-1500, 2009

[95] M.-J. Tsai, C.-S. Wang and J. Liu, "A Hybrid Model for Digital Camera Source Identification," *in Proc. of ICIP*, pp. 2901-2904, 2009

[96] L. Nataraj, A. Sarkar and B.S. Manjunath, "Adding Gaussian Noise to 'Denoise' JPEG for Detecting Image Resizing, " *in Proc. of ICIP*, pp. 1493-1496, 2009

[97] W. Wang, J. Dong and T. Tan, "Effective Image Splicing Detection Based on Image Chroma," *in Proc. of ICIP*, pp. 1257-1260, 2009

[98] X. Pan and S. Lyu, "Detecting Image Region Duplication Using SIFT Features," *in Proc. of ICASSP*, pp. 1706-1709, 2010

[99] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis Using Image Quality Metrics," *IEEE Trans. on Image Processing*, vol. 12(2), pp. 221-229, 2003

[100] Avcibas, M. Kharrazi, N. Memon, and B. Sankur, "Image Steganalysis with Binary Similarity Measures," *Journal of Applied Signal Processing*, vol. 17, pp. 2749-2757, 2005

[101] S. Lyu and H. Farid, "Steganalysis using Higher-Order Image Statistics," *IEEE Trans. on Information Forensics and Security*, vol. 1(1), pp. 111-119, 2006

[102] H. Farid, "Blind Inverse Gamma Correction," *IEEE Trans. on Image Processing*, vol. 10(10), pp. 1428-1433, 2001

[103] J. Lucas and J. Fridrich, "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images," *in Proc. of DFRWS*, 2003

[104] T.-T. Ng, S.-F. Chang, and M.-P. Tsui, "Using Geometry Invariants for Camera Response Function Estimation," *in Proc. of CVPR apos*, pp. 1-8, 2007

[105] A. Swaminathan, M. Wu, and K.J.R. Liu, "A Component Estimation Framework for Information Forensics," in Proc. of MMSP, pp. 397-400, 2007

[106] H. Yu, T.-T. Ng, and Q. Sun, "Recaptured Photo Detection Using Specularity Distribution," *in Proc. of ICIP*, pp. 3140-3143, 2008

[107] J. Mao, O. Bulan, G. Sharma and S. Datta, "Device Temporal Forensics: An Information Theoretic Approach," *in Proc. of ICIP*, pp. 1501-1504, 2009

[108] T.-T. Ng, S.-F. Chang, C.-Y. Lin, and Q. Sun, "Passive-Blind Image Forensics," *in Multimedia Security Technologies for Digital Rights*, 2006

[109] S.-F. Chang, "Blind Passive Media Forensics: Motivation and Opportunity," i*n Proc. of MCAM*, pp. 57-59, 2007

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

[111] H. Farid, "A Survey of Image Forgery Detection," *IEEE Signal Processing Magazine*, vol. 26(2), pp. 16-25, 2009

[112] A. Swaminathan, M. Wu and K.J.R. Liu, "Component Forensics," *IEEE Signal Processing Magazine*, vol. 26(2), pp.38-48, 2009

[113] J. Fridrich, "Digital image forensics," IEEE Signal Processing Magazine, vol. 26(2), pp. 26-37, 2009

[114] X. Jiang, B. Mandal, and A. C. Kot, "Eigenfeature Regularization and Extraction in Face Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30(3), pp. 383-394, 2008

[115] P. Pudil, F.J. Ferri, J. Novovicova, and J. Kittler, "Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions," *in Proc. of IAPR*, vol. 2, pp. 279-283, 1994

[116] M.-J. Tsai and C.-S. Wang, "Adaptive Feature Selection for Digital Camera Source Identification," *in Proc. of ISCAS*, pp. 412-415, 2008

[117] V.N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed.: Springer-Verlag, 1999

[118] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods," *Advances in Large Margin Classifiers (Neural Information Processing)*, pp. 61-74, 2000

[119] B. Scholkopf, J.C. Platt, J.S. Talor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," Neural Computation, vol. 13(7), pp. 1443-1471, 2001

[120] Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences*, vol. 55(1), pp. 119-139, 1997

[121] R.E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37(3), pp. 297-336, 1999

[122] S.Z. Li and Z. Zhang, "FloatBoost Learning and Statistical Face Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26(9), pp. 1112-1123, 2004

[123] P. Viola and M. Jones,"Robust Real-Time Face Detection," *Int'l J. Computer Vision*, vol. 57, pp. 137-154, 2004

[124] M. Kirchner, and R. Bohme, "Tamper Hiding: Defeat Image Forensics," *Lecture Notes in Computer Science*, vol. 4567/2008, pp. 326-341, 2008

[125] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can We Trust Digital Image Forensics?," *in Proc. Int. Conf. on Multimedia*, 2007

[126] M. Kirchner, and R. Bohme, "Synthesis of Color Filter Array Pattern in Digital Images," *in Proc. of SPIE*, vol. 7254, p. 72540K, 2009

[127] R. Steinman, "Digital Forensics An Interview with Dr. Hany Farid," The Digital Journalist, 2008, Available: http://digitaljournalist.org/issue0802/digital-forensics-an-interview-with-dr-hany-farid.html

[128] Foveon Official Website, Available: http://www.foveon.com/

[129] B.E. Bayer, "Color Imaging Array," *USA Patent*, 3,971,065, Eastman Kodak Company (Rochester, NY), 1976

[130] Digital Photography Review, Available: http://www.dpreview.com

[131] D.R. Cok, "Signal Processing Method and Apparatus for Sampled Image Signals," *USA Patent*, 4,630,307, Eastman Kodak Company (Rochester, NY), 1986

[132] D.R. Cok, "Signal Processing Method and Apparatus for Producing Interpolated Chrominance Values in a Sampled Color Image Signal," *USA Patent*, 4,642,678, Eastman Kodak Company (Rochester, NY), 1987

[133] W.T. Freeman, "Median Filter for Reconstructing Missing Color Samples," *USA Patent*, 4,724,395, Polaroid Corporation (Cambridge, MA) 1988

[134] C.A. Laroche and M.A. Prescoot, "Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients," *USA Patent*, 5,373,322, Eastman Kodak Company (Rochester, NY), 1994

[135] J.J.F. Hamilton and J.J.E. Adams, "Adaptive Color Plane Interpolation in Single-Sensor Color Electronic Camera," *USA Patent*, 5,629,734, Eastman Kodak Company (Rochester, NY), 1997

[136] R. Kimmel, "Demosaicing: Image Reconstruction from CCD Samples," *IEEE Trans. on Image Processing*, vol. 8(9), pp. 1221-1228, 1999

[137] X. Li and M.T. Orchard, "New Edge-Directed Interpolation," *IEEE Trans. on Image Processing*, vol. 10(10), pp. 1521-1527, 2001

[138] B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau, "Color Plane Interpolation Using Alternating Projections," *IEEE Trans. on Image Processing*, vol. 11(9), pp. 997-1013, 2002

[139] S.-C. Pei and I.-K. Tam, "Effective Color Interpolation in CCD Color Filter Arrays using Signal Correlation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13(6), pp. 503-513, 2003

[140] L. Chang and Y.P. Tan, "Effective Use of Spatial and Spectral Correlations for Color Filter Array Demosaicing," *IEEE Trans. on Consumer Electronics*, vol. 50(1), pp. 355-365, 2004

[141] D. Alleysson, S. Susstrunk, and J. Herault, "Linear Demosaicing Inspired by the Human Visual System," *IEEE Trans. on Image Processing*, vol. 14(4), pp. 439-449, 2005

[142] K. Hirakawa and T.W. Parks, "Adaptive Homogeneity-Directed Demosaicing Algorithm," *IEEE Trans. on Image Processing*, vol. 14(3), pp. 360-369, 2005

[143] X. Wang, W. Lin, and P. Xue, "Demosaicing with Improved Edge Direction Detection," *in Proc. of ISCAS*. vol. 3, pp. 2048-2051, 2005

[144] X. Li, B. Gunturk, and L. Zhang, "Image Demosaicing: a Systematic Survey," *in Proc. of SPIE*. vol. 6822, p. 68221J, 2008

[145] P.C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*: SIAM, 1998

[146] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2008

[147] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for Support Vector Machines," 2009

[148] L.I. Kuncheva, "A Theoretical Study on Six Classifier Fusion Strategies," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24(2), pp. 281-286, 2002

[149] “Rare-Tiger Photo Flap Makes Fur Fly in China,” Science, vol. 318, no. 5852, p. 893, 2007.

[150] “Beijing Fires Officials over Tiger Photo Hoax,” *in The Singapore Straits Times*, 30 June 2008

[151] F.-T. Pai, "Method for Driving a Liquid Crystal Display in a Dynamic Inversion Manner," USA Patent, 7,109,964, 2006

[152] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24(7), pp. 971-987, 2002