WARWICK
THE UNIVERSITY OF WARWICK

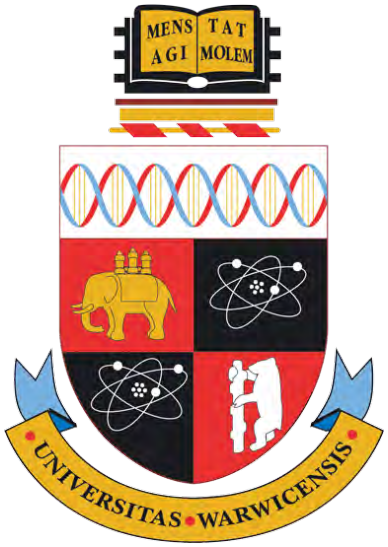


Photo Response Non-Uniformity Based Image Forensics in the Presence of Challenging Factors

by

Yijun Quan

Thesis

Submitted to the University of Warwick

in partial fulfilment of the requirements

for admission to the degree of

Doctor of Philosophy

Department of Computer Science

December 2020

Contents

List of Tables

List of Figures

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Chang-Tsun Li, who has provided me with tremendous support, kind advice and encouragement throughout my PhD studies. I am extremely grateful to my co-supervisor, Dr. Ligang He, for his valuable guidance and I am very thankful for my advisor, Dr. Victor Sahchez, for his insightful advice.

I would like to thank Dr. Xufeng Lin, both as a collaborator and a friend, who is always willing to lend a helpful hand, no matter of the circumstances. I am very grateful for Dr. Irene Amerini, Dr. Rahimeh Rouhi, Prof. Danilo Montesi and Dr. Xiangyu Yu, who gave me valuable advice and generous help on my PhD studies. I am blessed with all my lab mates, Dr. Bo Wang, Dr. Ning Jia, Dr. Qiang Zhang, Dr. Chao Chen, Dr. Roberto Leyva, Dr. Shan Lin, Dr. Haoyi Wang, Dr. Shenyuan Ren, Dr. Mohammed Alghamdi, Dr. Justin Chang, Junyu Li, Qingzhi Ma, Wentai Wu, Bowen Du, Yujue Zhou, Zhiyan Chen and Hao Wu. I always love the friendly and positive atmosphere in our lab and miss it so much after working from home for almost a year.

Many thanks to my friends, Danny Onsiong, Phoenix Tse, Liyun Ju, Yaxue Shen, Bonan Zhu, Junyang Huang, Guoshuai Cao, Jieyuan Wu, Hui Ding, Qiurui He, Lei Yuan, Yutian Wu, Chenglong Zhao, Tong Tong, and Valerian Hall-Chen. Their company, whether it is physical or virtual, has been the most precious, especially under this difficult time.

Lastly, I would like to express my deepest gratitude to my parents. They have always been my firmest support.

Declarations

Parts of this thesis have been previously published by the author in the following:

[1] Y. Quan and C.-T. Li, "On addressing the impact of ISO speed upon PRNU and forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 190–202, 2021

[2] Y. Quan, X. Lin, and C.-T. Li, "Provenance analysis for instagram photos," in *Australasian Conference on Data Mining*. Springer, 2018, pp. 372–383

[3] Y. Quan, C.-T. Li, Y. Zhou, and L. Li, "Warwick image forensics dataset for device fingerprinting in multimedia forensics," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6

[4] Y. Quan, X. Lin, and C.-T. Li, "Provenance inference for instagram photos through device fingerprinting," *IEEE Access*, vol. 8, pp. 168 309–168 320, 2020

Research was performed in collaboration during the development of this thesis, but does not form part of the thesis:

[5] R. Rouhi, F. Bertini, D. Montesi, X. Lin, Y. Quan, and C.-T. Li, "Hybrid clustering of shared images on social networks for digital forensics," *IEEE Access*, vol. 7, pp. 87 288–87 302, 2019

Abstract

With the ever-increasing prevalence of digital imaging devices and the rapid development of networks, the sharing of digital images becomes ubiquitous in our daily life. However, the pervasiveness of powerful image-editing tools also makes the digital images an easy target for malicious manipulations. Thus, to prevent people from falling victims to fake information and trace the criminal activities, digital image forensics methods like source camera identification, source oriented image clustering and image forgery detections have been developed.

Photo response non-uniformity (PRNU), which is an intrinsic sensor noise arises due to the pixels non-uniform response to the incident, has been used as a powerful tool for image device fingerprinting. The forensic community has developed a vast number of PRNU-based methods in different fields of digital image forensics. However, with the technology advancement in digital photography, the emergence of photo-sharing social networking sites, as well as the anti-forensics attacks targeting the PRNU, it brings new challenges to PRNU-based image forensics. For example, the performance of the existing forensic methods may deteriorate due to different camera exposure parameter settings and the efficacy of the PRNU-based methods can be directly challenged by image editing tools from social network sites or anti-forensics attacks. The objective of this thesis is to investigate and design effective methods to mitigate some of these challenges on PRNU-based image forensics.

We found that the camera exposure parameter settings, especially the camera sensitivity, which is commonly known by the name of the ISO speed, can influence the PRNU-based image forgery detection. Hence, we first construct the Warwick Image Forensics Dataset, which contains images taken with diverse

exposure parameter settings to facilitate further studies. To address the impact from ISO speed on PRNU-based image forgery detection, an ISO speed-specific correlation prediction process is proposed with a content-based ISO speed inference method to facilitate the process even if the ISO speed information is not available. We also propose a three-step framework to allow the PRNU-based source oriented clustering methods to perform successfully on Instagram images, despite some built-in image filters from Instagram may significantly distort PRNU. Additionally, for the binary classification of detecting whether an image's PRNU is attacked or not, we propose a generative adversarial network-based training strategy for a neural network-based classifier, which makes the classifier generalize better for images subject to unprecedented attacks.

The proposed methods are evaluated on public benchmarking datasets and our Warwick Image Forensics Dataset, which is released to the public as well. The experimental results validate the effectiveness of the methods proposed in this thesis.

Sponsorships and Grants

The work in this thesis is supported by the EU Horizon 2020 Marie Sklodowska-Curie Actions through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY).

Acronyms

AR Autoregressive model.

AUC-ROC Area Under ROC Curve.

BM3D Block-Matching and 3D filtering.

Bow Bag-of-Words.

CCD Charge-Coupled Device.

CDF Cumulative Distribution Function.

CFA Color Filter Array.

CFAR Constant False Acceptance Rate.

CINFISOS Content-based Inference of ISO Speeds.

CMOS Complementary Metal-Oxide-Semiconductor.

CNN Convolutional Neural Network.

CRF Camera Response Function.

DCT Discrete Cosine Transform.

DSLR Digital Single-Lens Reflex camera.

DSNU Dark Signal Non-Uniformity.

DWT Discrete Wavelet Transform.

EXIF EXchangeable Image File format.

FAR False Acceptance Rate.

FOV Field of View.

FPR False Positive Rate.

GAN Generative Adversarial Networks.

HDR High Dynamic Range.

IMA Instant Messaging Application.

ISO International Organization for Standardization.

JPEG Joint Photographic Experts Group.

LSE Least Square Estimator.

MCSC Multi-Class Spectral Clustering.

MRF Markov Random Field.

NC Normalized Cuts.

NCC Normalized Correlation Coefficient.

OLS Ordinary Least Squres regression.

PRNU Photo Response Non-Uniformity.

QE Quantum Efficiency.

ROC Receiver Operating Characteristic.

ROI Region of Interest.

SCI Source Camera Identification.

SDR Standard Dynamic Range.

SNN Shared Nearest Neighbours.

SNS Social Network Site.

SOC Source-Oriented Clustering.

SPN Sensor Pattern Noise.

SQ scalar quantization.

SVM Support Vector Machine.

TPR True Positive Rate.

WB White Balancing.

WEAC Weighted Evidence Accumulation Clustering.

Chapter 1

Introduction

1.1 Digital Image Forensics

'All warfare is based on deception.' — *The Art of War*

This famous quote from Sun Tzu not only depicts the wars on the physical battlefield but also the ones in the digital world between criminals and forensic investigators. Digital images are often viewed as a reflection of the real world. Their capability of precisely recording scenes makes them crucial evidence under many scenarios. Convinced by the images' realistic appearance, the viewers may take the integrity of the information conveyed for granted. However, the emergence of powerful image editing tools allows even unskilled people to easily manipulate the images, hiding or altering the content and metadata from the originals. Deceived by these images, the viewers may fall victim to malicious or criminal activities, and the forensic investigators could have more difficulty to trace the source of these activities. In order to win this digital 'warfare', the forensic investigators need to have a clear vision on these information.

The pervasiveness of maliciously edited images makes 'fact-check' the new normal in our daily life. Even images from some seemingly reliable sources may contain forgeries. In this year's United States presidential election campaign, the campaign team for the Republican candidate Donald John Trump was reportedly using a fake image of their opponent candidate Joseph Robinette Biden Jr. for social media advertisement[1]. The image (Figure 1.1) is forged by showing a fake earpiece worn by the Democratic candidate, indicating the symptoms of dementia and raising doubts about his ability of carrying presidential duties. By adding such a small object to an image, the misinformed voters may change their opinions about the candidates. This could potentially impact the election outcome and put American and global politics into a different shape.

[1] `https://www.forbes.com/sites/andrewsolender/2020/10/01/trump-ads-feature-biden-photo-edited-to-include-airpod-asking-whos-in-joes-ear/`

[2] Image source: `https://www.facebook.com/ads/library/?id=618906978778952`

Figure 1.1: An advertising image used by the Republican campaign team for the 2020 United States presidential election[2]. A fake earpiece is worn by the Democratic candidate Joseph Robinette Biden Jr.

While the previous example shows how a tampered image can significantly impact politics, genuine ones can also deceive people by having purposefully mislabelled sources. During the outbreak of SARS-CoV-2 in Italy earlier this year, a photo of patients treated outside the hospitals were circulated widely on the Internet (Figure 1.2). It was claimed that the hospitals had run out of beds due to the spread of the virus. However, the image was actually taken in Zagreb, the capital city of Croatia on 22nd March 2020, after an earthquake hit the country[3]. An image like this could spread misleading information to the general public and cause chaos on the health service when it has already taken heavy pressures from the pandemic.

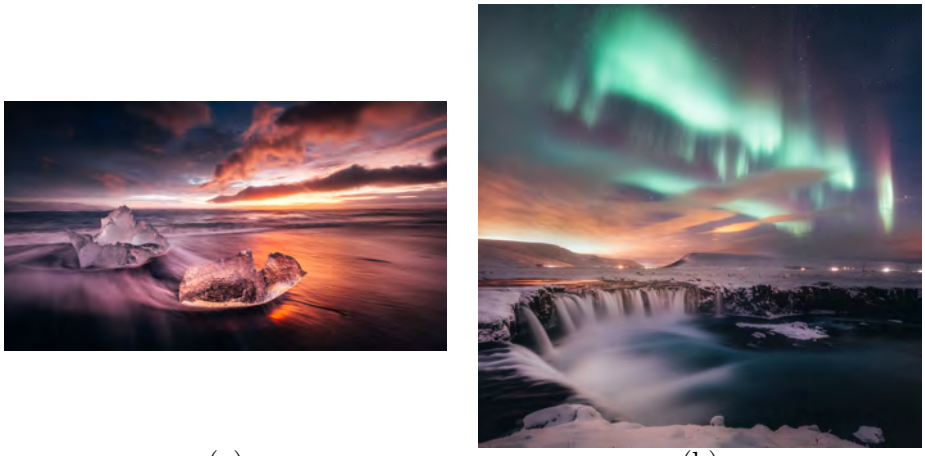
Falsely claimed image sources are often associated with unlawful economic gains as well. With the ubiquity of digital photography, the ability to take attractive photos has become a major selling point for many mobile devices. To promote their new devices, the phone manufacturer Huawei held a photography contest featuring images taken by their mobile devices. The two images shown in Figure 1.3 are told to be from a Huawei smartphone. The consumers are attracted by their aesthetic visuals, impressed with the camera's ability to shoot sharp images with high dynamic ranges. But by revealing the images' EXIF (Exchangeable image file format) file, the source of the two images is confirmed to be a Nikon D850 DSLR camera[4]. By identifying the source of the images, the consumers can correct their valuation of the device and not be deceived by the fraudulent advertisement.

[3] `https://www.bbc.co.uk/news/52124740`

[4] `https://www.scmp.com/abacus/tech/article/3080698/huawei-apologizes-using-dslr-shots-promote-smartphone-photo-contest`

[5] Image source: `https://500px.com/p/vcg-slayershute`

Figure 1.2: An image circulated on the Internet during the outbreak of SARS-CoV-2 in Italy, claiming the Italian hospitals ran out of beds and had to treat patients in the streets. However, the image is actually of the survivors to an earthquake in Croatia, March 2020.



(a) (b)

Figure 1.3: Two photos featured in a Huawei photography contest[5], which supposed to show images from Huawei devices only. The two images are claimed to be from a smartphone but later to be found out that their source is a Nikon D850 DSLR camera.

The above examples show us the potential damages by misleading information from untrustworthy digital images. Apart from that, digital images are often presented as evidence to law enforcement for investigation and jurisdiction processes. For crime investigation, revealing the underlying information regarding the history of an image would help the forensic investigators trace the criminals. For court jurisdiction, verifying the originality and integrity is necessary for the validity of an image to be used as evidence. Therefore, the importance and necessity of researching and developing forensic technologies to

verify images' originality, integrity, and authenticity are widely acknowledged.

With these in mind, the digital image forensic community wish to develop methods to achieve goals including but not limited to the ones shown below:

- Camera Model Identification: Given an image and different models of cameras (e.g. Canon 6D, Nikon D7200, and etc.), identify the model of the image's source camera.
- Source Camera Verification: Given an image and a device, verifying whether the image is captured by the device.
- Source Camera Identification: Given an image and a set of cameras, identify the specific camera if the image is captured by one of the cameras from the set.
- Source-Oriented Clustering: Given a set of images, sort the images to groups according to their source devices.
- Image Forgery Detection and Localization: Given an image, determine whether it contains forged pixels and locate them.
- Image History Retrieval: Given an image, retrieving its processing history and its online sharing footprint.

1.1.1 Active Digital Image Forensics

Active digital image forensics approaches are implemented by actively adding information about the images' originality, integrity and authenticity to images. Similar to the artists putting their signatures or special symbols to remark the origins of the artworks ever since the time before the Renaissance, inserting digital watermark [6–11] and signature [12–15] to digital images when they are captured are a straightforward method to determine the originality, verify the integrity and analyse the owner authenticity of the images. Despite the effectiveness of these active methods shown in the literature, the forensic investigators can only apply them by knowing that the digital watermark or signature was introduced to an image right after the image acquisition process. However, digital watermark and signature face several problems and restrictions that make the camera manufacturers reluctant to embed them in every image. Firstly, as mentioned above, the digital watermark or signature has to be introduced right after the image acquisition process, meaning the camera manufacturer has to add extra steps in the image processing pipeline. As the manufacturers usually do not benefit from these complementary functions, not every manufacturer has the incentive to introduce these extra steps in their camera pipeline. Secondly, the embedded watermark or signature may

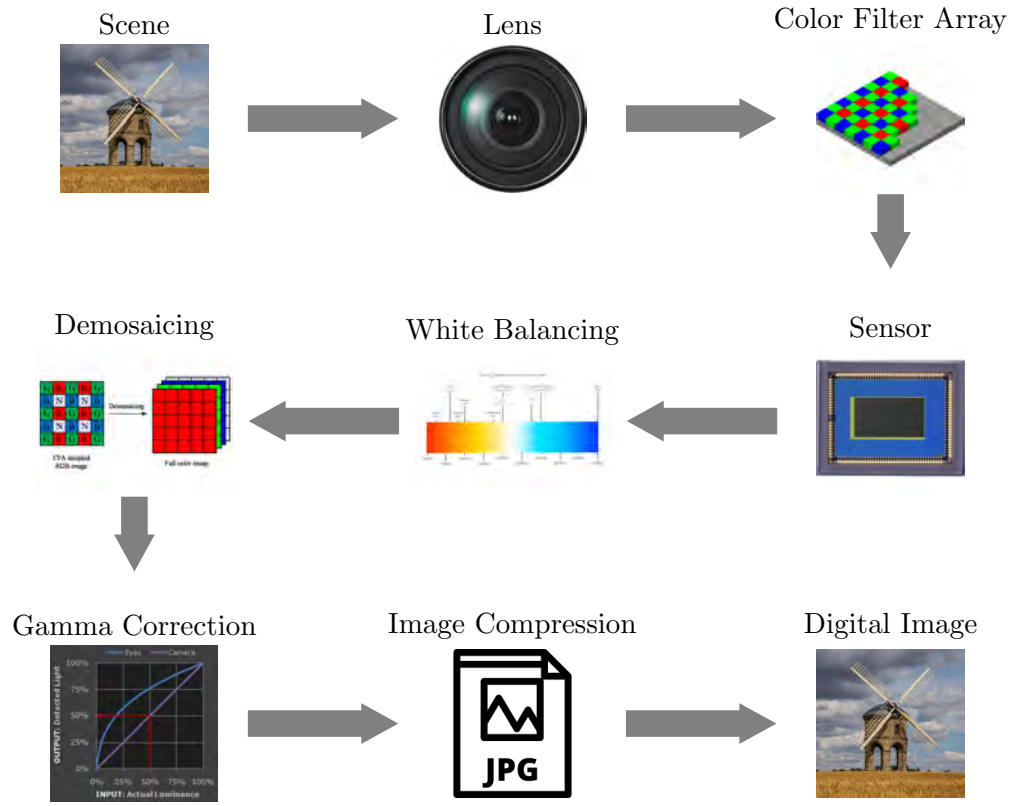


Figure 1.4: The typical image acquisition and storage pipeline of a digital camera

downgrade the image quality and hence, camera manufacturers may tend not to use them to keep their cameras' competitiveness in the market. With the aforementioned problems and restrictions on these active methods, the forensic research community has to look for practical passive digital forensic methods as well.

1.1.2 Passive Digital Image Forensics

Compared with active digital image forensic methods, passive methods do not require any prior or present information manually added to images. Instead, passive methods look for certain intrinsic patterns embedded in the images. These patterns can be considered as traces of different processes or manipulations during different stages in the image processing and editing pipeline. In general, methods are developed based on three types of traces [16]:

1. **Traces left during image acquisition**:
   There are several steps in the image acquisition process and different components of a camera could leave unique artifacts and patterns in an image which could be used for digital image forensics. Figure 1.4 shows the typical image acquisition pipeline of a digital camera. The light of a

scene is first captured by the lens of a camera. As a lens usually does not have perfect optical performance due to the material and manufacturing, it could introduce two types of lens aberrations to an image: chromatic aberration and spherical aberration. These lens aberrations can be used as traces to identify images taken by a particular lens and exposing image forgery [17, 18]. After passing through the lens, the light will be filtered by the color filter array (CFA) into different color channels before it is collected by the image sensor. The image sensor converts the light signal to electrical signal with a camera model-dependent quantum efficiency (QE) and in this process, it could exhibit distinctive noise characteristics and a sensor pattern noise can be introduced to the image. The noise characteristics of a sensor could be used for camera model identification [19] while the sensor pattern noise is a more powerful tool which can be used for more specific source device identification and has been used in a broader area of digital image forensics as well [20–26]. With the light captured by the sensor, the camera will perform white balancing (WB) on the image to adjust its color temperature. Remembering the light was filtered by CFA, the signal on each pixel at this stage only accounts for one color channel. Thus, to create a color image, CFA demosaicing has to be done following the white balancing (note that there are also some camera manufacturers implementing CFA demosaicing before white balancing). The demosaicing process interpolates the neighbouring pixels' intensity and as a result, artifacts are introduced. CFA demosaicing artifacts could be used for camera model identification [27, 28] and image forgery detection [29–31]. After the demosaiced channels being fused together to form the color image, gamma correction will be applied. The gamma correction translates the irradiance received on the sensor to the actual brightness of the displayed image. Together with the aforementioned quantum efficiency, these two terms generally define the camera's overall response to the incident light's luminance and can be formulated as a camera response function (CRF). Various forensic methods [32–35] have been developed based on the characteristics of the camera response function. After gamma correction, the image will be compressed and stored. These processes contribute to another group of traces left in images which can be used for passive forensics.

2. **Traces left in image storage and distribution:**
   With people's increasing need for images with better details and the development of digital photography, the resolution of a modern digital image is usually measured in the unit of million pixels. Without any compression, the storage and distribution for color images of such a big

size become rather infeasible. Thus, different compression standards are used. JPEG is one of the most widely used formats for image compression. This lossy compression leaves artifacts in images. By studying the pattern of these artifacts in an image, some information about the image's processing history could be revealed, e.g. how many times the image has been compressed and whether all the regions of the image have been compressed for the same number of times. Information like these gives clues about the existence of the tampered region in an image as for most tampered images, at least two JPEG compressions have been applied to the original images. Hence, many literature focus on the study of detecting double JPEG compression and forgery detection methods based on it [36–45]. With the emergence of social network sites (SNS) and instant messaging applications (IMA), the fast transmission of images via the Internet and the storage of the vast number of images on the sites' servers require these service providers to further compress the images. Each SNS or IMA could apply platform-specific manipulations during the compression process, leaving unique traces that can reveal the image online-sharing history [46–49].

3. **Traces left by image editing:**
   When an attacker tries to forge an image, they will have to apply one or more image manipulations to edit the image. Each type of image manipulation could leave some specific traces in the attacked image. Thus, corresponding forensic methods could be developed by exploiting these manipulation-specific traces. These manipulations include median filtering [50–57], unsharp masking [58–60], resampling [61–63] and copy-move attacks [64–68]. In addition, contrast enhancement [69–72] and inconsistent lighting [73–76] introduced by image editing tools could also be used in image forgery detection.

1.2 Photo Response Non-Uniformity Based Image Forensics

While different camera artifacts and manipulation traces are used for passive image forensic, among them, sensor pattern noise (SPN) has shown its strength. SPN has the following properties which make it a powerful tool for digital image forensics:

1. SPN is unique to each sensor.

2. SPN is stable against environmental conditions.

3. SPN is a pixel-level signal presented in the whole image which makes it suitable for both pixel-level (e.g. image forgery localization) and image-level (e.g. source camera identification and source-oriented clustering) applications.

There are two major components of SPN: dark signal non-uniformity (DSNU) and photo response non-uniformity (PRNU). The DSNU is the pixels' non-uniform response to the environment when the sensor is not exposed to light. Thus, it is also know by the name of dark current noise. This signal is relatively weak in images under normal lighting conditions and many camera manufacturers provide calibration function to attenuate this signal. As a result, it is difficult to build reliable forensic methods based on DSNU. In comparison, PRNU becomes the physical foundation for SPN based forensic methods.

PRNU mainly arises due to the manufacturing imperfections of the silicon wafers used to build the image sensors. An image sensor, no matter whether it is a charge-coupled device (CCD) or a complementary metal-oxide-semiconductor (CMOS) sensor, will have slightly different quantum efficiency (QE) on each pixel due to the inhomogeneity introduced by the manufacturing imperfections. The quantum efficiency defines the pixel's ability to convert the light signal to the electrical signal. Consequently, the pixels on a sensor will have a non-uniform response to the incident light. Hence, this phenomenon gives the name to PRNU. PRNU can be viewed as the fingerprint of an imaging device and is applied in different fields of digital image forensics as will be briefly explained in the following subsections.

1.2.1 PRNU-based Source Camera Identification

Source camera identification is the task of identifying an image's source device given a set of candidate cameras. The procedure of a typical PRNU-based source camera identification framework is shown in Figure 1.5. With a query image, its PRNU could be estimated by the noise extracted from it. For each candidate camera, a reference PRNU could be constructed from the noises extracted from multiple reference images, often by simply taking the average of the extracted noises. To maximally avoid the interference from image scenes, usually flatfield images (e.g., blue sky or pure color images) are used for high quality reference PRNU extraction. With the reference PRNUs extracted, the similarity between the query PRNU and each camera's reference PRNU could be calculated, often using normalized cross correlation as a measurement. After that, the source camera could be identified as the one that has the highest similarity with the query PRNU, if this similarity measurement is higher than a predefined threshold.

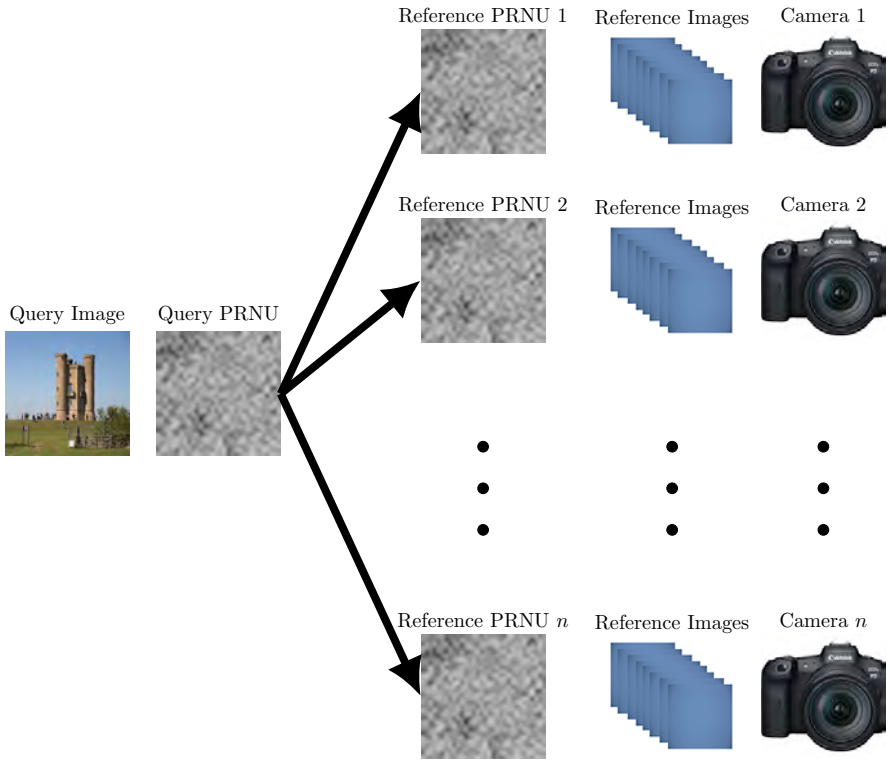


Figure 1.5: The procedure of a typical PRNU-based source camera identification framework.

1.2.2 PRNU-based Source-Oriented Clustering

Source-oriented clustering considers the scenario of having multiple images with unknown sources as the input and grouping them according to common source devices. It can help the forensic investigators to understand the underlying connections between images. PRNU-based clustering exploits the similarity between the images' extracted PRNUs. The same as it is done in the PRNU based source camera identification, each image's PRNU could be estimated by its extracted noise. The similarity between image pairs could be measured by the pairwise cross-correlations between all the images. After the pairwise similarities are computed, by grouping PRNUs with high similarities together, the corresponding images can be clustered into the same groups according to their source devices. An illustration is shown in Figure 1.6.

1.2.3 PRNU-based Image Forgery Localization

With PRNU being a pixel-level signal, it allows a localised detection of PRNU's presence in an image. This could reveal the location of tampered pixels. The localised detection is usually done in a block-wise manner due to the weak nature of PRNU which requires a relatively large number of pixels to deliver a

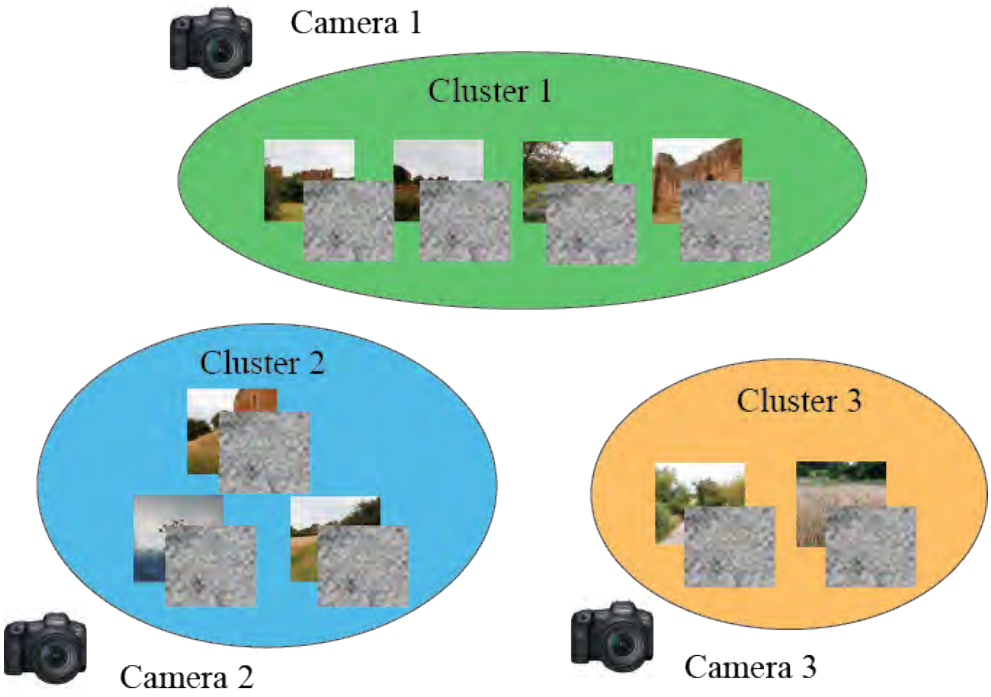


Figure 1.6: An illustration of PRNU-based source-oriented image clustering.

reliable detection. A correlation map can be computed between the estimated PRNU extracted from the image and the camera's reference PRNU by shifting the computation block over the whole image. Again, due to the weak nature of PRNU, the correlation map itself hardly indicates the exact location of tampered regions. The correlation map has to be compared with a prediction map, which needs to be constructed using a feature-based predictor, to generate the detection result. Figure 1.7 demonstrates the pipeline of the PRNU-based image forgery localization.

1.3 Main Contributions

While PRNU-based image forensics has been a well-studied research area with a lot of well-established forensic techniques, new challenges are faced due to the recent development in digital photography. This thesis identifies some of these challenges and proposes methods to tackle them. The major contributions made in this thesis are summarised in detail as follows.

1. Over the past few years, the rapid advancement in digital photography has greatly reshaped the pipeline of image capturing process on consumer-level imaging devices. The flexibility of camera parameter settings and the emergence of multi-frame photography algorithms, especially high dynamic range (HDR) imaging, bring new challenges to PRNU-based

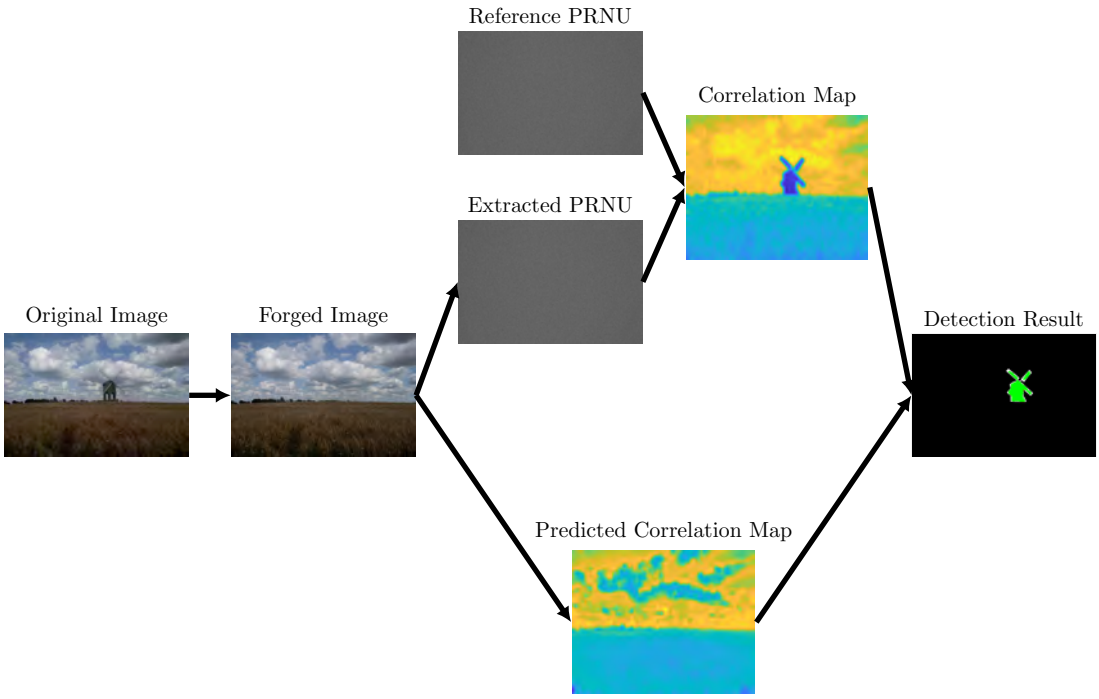


Figure 1.7: A demonstration of PRNU-based image forgery localization.

image forensics. We identify these challenges and acknowledge the need for subsequent studies on these topics. To facilitate these studies, we introduce a new purposefully built image dataset in Chapter 3, namely the Warwick Image Forensics Dataset. The dataset contains more than 58,600 images captured using 14 digital cameras with various exposure settings. Special attention to the exposure settings allows studies to be done on exposure parameters' impact on PRNU-based image forensics as shown in Chapter 4. Besides, this feature also makes it easy for the images to be adopted by different multi-frame computational photography algorithms. Despite this thesis does not include direct quantitative investigations on multi-frame computational photography algorithms' impact on PRNU-based image forensics, this dataset provides a platform for future work to be carried out.

2. The Warwick Image Forensics Dataset allows more dedicated and in-depth investigations into camera sensitivity settings' impact on PRNU-based image forgery detection. Camera sensitivity is more commonly known by the name of ISO speed by photographers. It determines the signal gain in the image acquisition process. In Chapter 4, we first derived a theoretical model for PRNU's relative strength in an image using a Poissonian-Gaussian noise model. It shows the dependency of the PRNU's strength on the camera's ISO speed. As shown in Figure 1.7, the localization of image forgery requires an accurate correlation map

prediction. Due to the dependency of PRNU's strength on the ISO speed, we further show how the ISO speed could impact the correlation prediction process. We propose an ISO-specific correlation prediction process and a Content-based Inference of ISO speed (CINFISOS) method to address this problem.

3. The emergence of photo-sharing social networking sites (SNSs) also poses new challenges to PRNU-based digital forensics. In Chapter 5, we identify one particular issue that the built-in image editing tools from SNSs could inflict distortion on PRNUs. One well-known example of such tools is the image filters on Instagram. We observed that some Instagram image filters manipulate the high-frequency bands of the images and hence damage the PRNUs, making source-oriented clustering of the filtered images unsatisfactory. To address this issue, we propose a three-step clustering framework by separating the images processed by different filters into two groups. To identify the filter applied to each image, a convolutional neural network (CNN) based filter-oriented image classifier is proposed. By treating the two groups of images separately, the proposed framework manages to cluster Instagram images despite the heavy distortion of PRNUs from certain filters.

4. Anti-forensics attacks can be considered as the more direct challenges to PRNU-based techniques. Being a noise-like signal, PRNU could be attenuated or removed by some simple manipulations like median filtering or Gaussian blurring. Thus, corresponding counter anti-forensics methods are required. Recent development in neural network-based methods has seen successes in extracting features for different anti-forensics manipulations[77–79]. However, neural network classifier trained using images attacked by a specific group of manipulations do not generalise well for other unprecedented attacks. Thus, the aforementioned networks can perform well on detecting specific attacks but not for the more general task: *the binary classification of detecting whether the PRNU in an image is attacked or not.* To help a neural network generalise better for this task despite the limitations of the training images, in Chapter 6 we propose a training strategy using generative adversarial networks (GAN). This training strategy can prevent the classifier from putting excessive emphasis on the manipulation-specific features and the resultant classifier can generalise better for unprecedented anti-forensics attacks.

1.4 Outline of Thesis

This chapter briefly introduces the background of digital image forensics and the application of PRNU-based methods in source camera identification, source-oriented image clustering, and image forgery detection. The rest of this thesis is presented in the following structure.

Chapter 2 first reviews the PRNU-based image forgery detection with an emphasis on correlation prediction and its role in different forgery detection algorithms. It then revisits the Poissonian-Gaussian noise model built for the raw image capturing process on digital cameras. After that, different source-oriented image clustering algorithms are reviewed. The last section of this chapter reviews the literatures on different counter anti-forensics attack methods.

Chapter 3 first discusses how camera parameter settings and multi-frame merging algorithms may impact PRNU-based image forensics. By showing the limitations of the existing public image forensics datasets, the underlying design of the Warwick Image Forensics Dataset is then presented in this chapter. After evaluating the dataset using benchmarking source camera identification tests, this chapter concludes with a discussion of how this dataset could help research on PRNU-based forensics to be carried out.

Chapter 4 starts with a further development of the theoretical Poissonian-Gaussian noise model reviewed in Chapter 2 by including the PRNU factor. This further development of the model analytically proves that a camera's ISO speed may impact PRNU's strength in flatfield images. In addition to the studies on the flatfield images, this chapter empirically shows that this impact is also presented in more general cases and may affect the correlation prediction process in PRNU-based forgery detection. To address this problem, this chapter proposes an ISO speed-specific correlation prediction process followed by a Content-based Inference of ISO speed (CINFISOS) algorithm.

Chapter 5 first investigates the impact of image editing tools used by social network sites on PRNU-based source camera identification and source-oriented image clustering. Using Instagram filters as an example, we show the impact from the image filters on PRNU-based source-oriented image clustering. Hence, a three-step image clustering framework is proposed in Chapter 5 to allow PRNU-based source oriented image clustering to perform on Instagram images. A convolutional neural network-based image filter classifier is also presented in this chapter.

Chapter 6 identifies the limitations of the existing neural network based anti-forensics attack detection methods. These methods may put excessive emphasis on manipulation-specific features and do not generalise well for unprecedented anti-forensics attacks. A generative adversarial networks-based

training framework is proposed in this chapter. It makes the neural-network based classifiers generalise better for the binary classification of detecting whether the PRNU in an image is attacked or not, despite the training set may only contain certain types of attacks. The effectiveness of the proposed framework is tested and demonstrated in the improved accuracy in identifying images attacked by three different manipulations unprecedented from the training set.

Chapter 7 concludes this thesis. A summary of the challenges we tackled and possible future research directions are given in this chapter.

Chapter 2

Literature Review

This chapter will review some of the existing PRNU-based forensic methods and point out the challenges they face. Firstly, Section 2.1 reviews the PRNU-based image forgery detection methods. We will show how the PRNU correlation predictor is formulated in [23] and its usage in different forgery detection methods. Following that, Section 2.2 will review a Poissonian-Gaussian noise model which describes the image capturing process for raw images with digital cameras. By gaining an insight into this noise model, a solid foundation could be built for detailed investigations into ISO speed's impact on the PRNU correlation predictor. A method to estimate the expectations and the variances of pixels from noisy images, which will facilitate our studies on camera noise model, will also be revised. Different source-oriented clustering methods will be discussed in Section 2.3. Section 2.4.2 discusses anti-forensics attacks on PRNUs and the corresponding countering and detection methods for these attacks. Details on how convolutional neural networks can be used for attacks on PRNU will be presented.

2.1 PRNU-based Image Forgery Detection

2.1.1 Preliminary Method

PRNU-based image forgery detection is first proposed in [21], in which the authors address the forgery detection problem by verifying the integrity of a selected Region of Interest (ROI). Two different approaches were devised in [21]. To conduct PRNU-based forgery detection, first, both the PRNU from the image in question and the reference PRNU from the camera have to be extracted and estimated. To construct the reference PRNU, $\mathbf{R}$, of the camera, C, by which the image in question $\mathbf{Y}$ is taken, the average of the noise residuals

$\mathbf{W}$ from N reference images from C is calculated:

$$\mathbf{R} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{W}_i \tag{2.1}$$

The noise residual $\mathbf{W}$ of an image $\mathbf{I}$ is formulated as:

$$\mathbf{W} = \mathbf{I} - F(\mathbf{I}), \tag{2.2}$$

where $F(\mathbf{I})$ denotes the denoised version of an image $\mathbf{I}$. By following Equation (2.2), the PRNU for $\mathbf{Y}$ could be estimated as its noise residual $\mathbf{Z}$.

The first approach from [21] is proposed to verify the integrity of a selected area Ω in an image $\mathbf{Y}$. The similarity between the reference PRNU $\mathbf{R}$ and the noise residual $\mathbf{Z}$ in the selected region Ω can be measured by the normalized correlation coefficient (NCC):

$$\rho(\mathbf{R}_\Omega, \mathbf{Z}_\Omega) = \frac{\sum_{q\in\Omega}(\mathbf{R}[q] - \bar{R})(\mathbf{Z}[q] - \bar{Z})}{\|\mathbf{R} - \bar{R}\| \cdot \|\mathbf{Z} - \bar{Z}\|}, \tag{2.3}$$

where q denotes a pixel in the region Ω and $\|\cdot\|$ is the L_2 norm. $\bar{R}$ and $\bar{Z}$ are the arithmetic means of $\mathbf{R}$ and $\mathbf{Z}$, respectively.

As PRNU is a weak noise, the correlation between the reference and the noise residual is usually small even for pristine images. Thus, to better differentiate pristine and tampered regions, the prior knowledge about the expected correlation distribution of the region if it is tampered is required. To calculate this statistics, a large set of L image regions $\mathbf{Q}_k, k = 1, \ldots, L$ of the same size and shape is collected either from the images taken by the same camera C but a different location within the images or from the images taken by other cameras. As these regions do not share the same PRNU with the reference PRNU, they can be considered as 'tampered'. In addition, as the *inter-class correlation* (*i.e.* the correlation between PRNUs of different sources) is not content-dependent, the collected correlations follow the same expected distribution despite the regions depicting different content. Thus, a generalised Gaussian distribution can be estimated from the correlations, $\rho(\mathbf{R}_\Omega, \mathbf{W}_k), k = 1, \ldots, L$, between these regions' noise residuals $\mathbf{W}_k, k = 1, \ldots, L$ with the reference PRNU, R_Ω, in the region Ω. Using the estimated generalised Gaussian distribution, the probability, p, of a tampered region with correlation bigger than $\rho(\mathbf{R}_\Omega, \mathbf{Z}_\Omega)$ can be given as:

$$p = 1 - G(\rho(\mathbf{R}_\Omega), \mathbf{W}_\Omega)) \tag{2.4}$$

where $G(\cdot)$ is the cumulative distribution function (CDF) of the estimated generalised Gaussian distribution. With this probability, the decision statistics

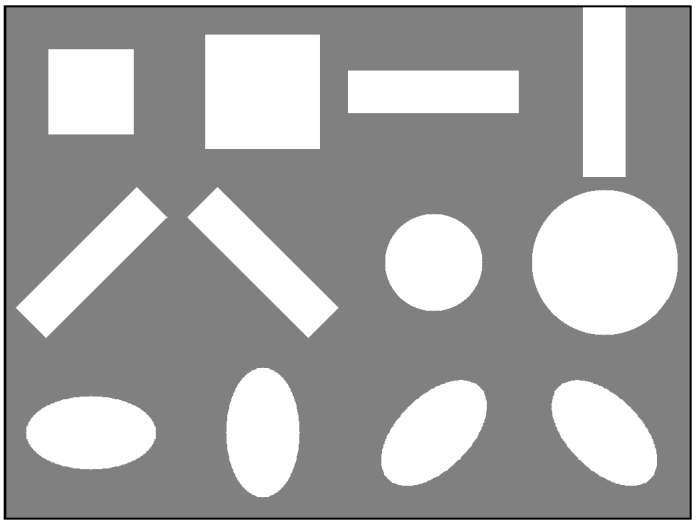


Figure 2.1: The 12 different sliding blocks used for ROI detection. The size of the sliding blocks are defined in [21].

could be obtained by setting a constant threshold: Ω has been forged if $p > \alpha$ and not forged otherwise with $\alpha = 10^{-3}$ in [21].

The second approach from [21] is able to identify ROI automatically. To detect forgeries of different shapes, [21] uses twelve sliding blocks of different shapes and sizes as shown in Figure 2.1. These blocks will slide across the entire image and the correlation within each block will be calculated for decision making. This method can be summarised into the following 5 steps:

1. For each block type i from the N types of blocks shown in Figure. 2.1, compute the correlations between the blocks' noise residual with the camera's reference PRNU over the whole image in a sliding window manner. This will generate a number of correlations $\rho_j, j = 1, \ldots, n_i$, where n_i is the number of the correlations computed for the ith block type.

2. For each block type i, select m blocks with the smallest correlations $\rho_k, k = 1, \ldots, m$ to form a set $\mathfrak{B}_k$. There will be a total of $m \times N$ blocks in $\mathfrak{B}_k$

3. Combine all the selected blocks and construct the mask $\mathfrak{B} = \cup_{k=1}^{m \times N} \mathfrak{B}_k$.

4. For each pixel $q \in \mathfrak{B}$, count the number of blocks selected in Step 2 with q included: $t(q) = |\{\mathfrak{B}_k | q \in \mathfrak{B}_k\}|$.

5. The pixels in ROI are the ones with $t(q)$ bigger than the median value T of $t(q)$ for $q \in \mathfrak{B}$: $\mathfrak{R} = \{q | t(q) > T\}$

This method can automatically identify ROI and the identified ROI can be further verified using the first approach from [21]. The two methods from [21]

are the earliest attempts of using PRNU for image forgery detection and are further extended by [23] as shown in the following sections.

2.1.2 PRNU Correlation Prediction

The preliminary methods from [21] show the possibility of using PRNU for image forgery detection. But the use of the generalised Gaussian distribution of correlations from tampered regions with the threshold α suggests that the method expects a constant false acceptance rate (FAR, the rate of identifying tampered pixels as pristine) without considering the false positive rate (FPR, the rate of misjudging pristine pixels as tampered). With the weak nature of PRNU, the distribution of correlations for the pristine regions could have large overlapping with the tampered regions' correlation distribution. Hence, it is not uncommon to witness false positives, which limits the applicability of PRNU-based forgery detection. Thus, a reliable forgery detection algorithm needs to take the correlation distribution of the pristine regions into the consideration. Most existing PRNU-based image forgery detection methods analyse the correlation distribution of the pristine regions via a correlation predictor proposed in [23]. We use this subsection to present its details.

By dividing an image into blocks, [23] treats the PRNU-based forgery detection as a binary hypothesis testing problem in each block:

$$\begin{cases} H_0 : \mathbf{W} = \mathbf{\Xi}, \\ H_1 : \mathbf{W} = \mathbf{R} + \mathbf{\Xi}, \end{cases} \tag{2.5}$$

where $\mathbf{W}$ is the noise residual extracted from the image in question, $\mathbf{I}$. $\mathbf{R}$ is the reference PRNU of the source camera C. $\mathbf{\Xi}$ denotes the PRNU-irrelevant noise in the noise residual. H_0 is the hypothesis that the noise residual comes from a tampered block with no existence of the source camera's PRNU. H_1 considers the extracted noise residual as containing both PRNU component and other irrelevant noise. With a correlation x calculated from the block following Equation (2.3), the probability of the block conforming with hypothesis H_0, $p(x|H_0)$ can be estimated in the same way as it is done in Equation (2.4) from [21]. But to compute the probability of having correlation x under hypothesis H_1 requires an expected correlation for pristine block to be predicted.

The prediction of correlation distribution for the pristine blocks is not as straightforward as the calculation of the distribution for the tampered blocks. From the form of the pristine noise residual expressed in Equation (2.5), it is trivial that the correlation would depend on the relative strength of the PRNU component $\mathbf{R}$ in $\mathbf{W}$ compared to the PRNU-irrelevant part $\mathbf{\Xi}$. By analysing the noise residual components, [23] proposed a correlation predictor for the pristine images as an image feature-based model.

To find the factors which could affect the relative strength between $\mathbf{R}$ and $\mathbf{\Xi}$, and thus the PRNU correlation, [23] considers the following sensor output model for an image $\mathbf{I}$:

$$\mathbf{I} = g^{\gamma} \cdot [(\mathbf{1} + \mathbf{K})\mathbf{Y} + \mathbf{\Lambda}]^{\gamma} + \mathbf{\Theta}_q, \tag{2.6}$$

where g is the camera gain, γ is the coefficient for gamma correction. $\mathbf{Y}$ is the scene light intensity and $\mathbf{K}$ denotes the pixel's non-uniform response to the light, thus representing the PRNU factor. $\mathbf{\Lambda}$ is a combination of the other noise sources including the dark current, shot noise, and read-out noise. $\mathbf{\Theta}_q$ is the quantization noise. As in natural images, the image signal is more dominant than the noise. This allows the Taylor expansion to be applied for the equation and an approximation can be made by only keeping the lower order terms. It gives

$$\mathbf{I} \approx \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta}. \tag{2.7}$$

The signal $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$ is the noise-free sensor output. $\mathbf{I}^{(0)}\mathbf{K}$ denotes the PRNU term in the output signal and $\mathbf{\Theta} = \gamma\mathbf{I}^{(0)}\mathbf{\Lambda}/\mathbf{Y} + \mathbf{\Theta}_q$ is a complex of independent random noise components.

To extract the noise residual $\mathbf{W}$ from $\mathbf{I}$, we use Equation (2.2):

$$\mathbf{W} = \mathbf{I} - F(\mathbf{I}) = \mathbf{I}\mathbf{K} + \mathbf{I}^{(0)} - F(\mathbf{I}) + (F(\mathbf{I}) - \mathbf{I})\mathbf{K} + \mathbf{\Theta} = \mathbf{I}\mathbf{K} + \mathbf{\Xi} \tag{2.8}$$

This expression shows that the PRNU term, $\mathbf{IK}$, in the noise residual is multiplicative with the image intensity. Thus, the first image feature considered by [23] is image intensity f_I. Within a block, B_b, the feature is defined as:

$$f_I = \frac{1}{|B_b|} \sum_{i \in B_b} \mathrm{att}(\mathbf{I}[i]), \tag{2.9}$$

where $|B_b|$ is the size of the block and $\mathrm{att}(\mathbf{I}[i])$ is the attenuated pixel intensity at pixel i:

$$\mathrm{att}(\mathbf{I}[i]) = \begin{cases} e^{-(\mathbf{I}[i] - I_{\mathrm{crit}})^2/\tau}, & \mathbf{I}[i] > I_{\mathrm{crit}}, \\ \mathbf{I}[i]/I_{\mathrm{crit}}, & \mathbf{I}[i] \leq I_{\mathrm{crit}} \end{cases} \tag{2.10}$$

The attenuation function is design to account for the clipping effect on the PRNU term when the pixel is too dark or saturated. The critical intensity I_{crit} and the attenuation factor τ is set to be 250 and 6 empirically in [23] for 8-bit images.

The form of $\mathbf{\Xi}$ in Equation (2.8) involves the difference between the noise-free sensor output $\mathbf{I}^{(0)}$ and the denoised image $F(\mathbf{I})$. As it is almost impossible to find a perfect denoising algorithm, the highly textured image components may propagate into $\mathbf{\Xi}$. Hence, the second image feature [23] considers is the

image texture, f_T:

$$f_T = \frac{1}{B_b} \sum_{i \in B_b} \frac{1}{1 + \text{var}_5(\mathbf{F}[i])}, \tag{2.11}$$

where $\mathbf{F}$ is the high-pass-filtered version of the image $\mathbf{I}$. $\text{var}_5(\mathbf{F}[i])$ computes the variance of $\mathbf{F}$ in the 5×5 neighbourhood of pixel i.

The authors of [23] also identifies that the image intensity and texture could collectively influence the the PRNU correlation. Thus, a texture-intensity combined feature, f_{TI}, is also included in the correlation predictor:

$$f_{TI} = \frac{1}{B_b} \sum_{i \in B_b} \frac{\text{att}(\mathbf{I}[i])}{1 + \text{var}_5(\mathbf{F}[i])}. \tag{2.12}$$

Another image feature identified by [23] is signal flattening, f_S. This feature accounts for the attenuation on PRNU by low-pass filtering operation like JPEG compression. f_S is defined with respect to the ratio of pixels with standard deviation in their local 5×5 neighbourhoods smaller than an intensity-dependent threshold:

$$f_S = \frac{1}{B_b} |\{i \in B_b | \sigma_I[i] < c\mathbf{I}[i]\}|, \tag{2.13}$$

where $\sigma_I[i]$ is the standard deviation in pixel intensity of the 5×5 neighbourhood around pixel i and c is a constant and set to 0.03 in [23].

With the four image features defined, [23] models the correlation predictor as a linear combination of the four features and their second order terms. For a pixel k, the PRNU correlation within the block B_b around i is formulated as:

$$\begin{aligned} \rho[k] = \theta_0 + \theta_1 f_I[k] + \theta_2 f_T[k] + \theta_3 f_S[k] + \theta_4 f_{TI}[k] + \\ \theta_5 f_I[k] f_I[k] + \cdots + \theta_{14} f_{TI}[k] f_{TI}[k] + \Psi[k], \end{aligned} \tag{2.14}$$

where $\Psi[k]$ is the modelling noise and θ is the coefficients to be determined. Considering we have K image blocks from the same camera and their computed PRNU correlation, we can rewrite Equation (2.14) into a matrix form: $\rho = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\Psi}$ with $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{14})$. As there are in total 15 terms in Equation (2.14) (1 zeroth order term, 4 first order terms and 10 second order terms), $\mathbf{H}$ is a $K \times 15$ matrix with each entry can be computed following the definition of the features and their combinations from the K image blocks. By applying the least square estimator (LSE), the parameters can be estimated as:

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\rho}. \tag{2.15}$$

Given an image block, its expected correlation $\hat{\rho}$ can be predicted as:

$$\hat{\rho} = [1, f_I, f_T, f_S, f_{TI}, \ldots, f_{TI} f_{TI}] \hat{\boldsymbol{\theta}} \tag{2.16}$$

With Equation (2.16) formulating the expected correlation for an image block, the correlation distribution under hypothesis H_1 can be modelled as the generalised Gaussian distribution $GG(\hat{\rho}, \sigma_1, \alpha_1)$, where the mean is the predicted correlation $\hat{\rho}$. The scale parameter σ_1 and shape parameter α_1 can be estimated by fitting the generalised Gaussian model $GG(0, \sigma_1, \alpha_1)$ to the prediction error $\nu = \rho - \hat{\rho}$ for all the image blocks used in training of $\hat{\boldsymbol{\theta}}$.

2.1.3 Constant False Acceptance Rate Method

By modelling the distribution of the PRNU correlation, the forgery detection can be performed in a pixel-wise manner over the whole image using a sliding window (e.g., a window of size 128×128 pixels), to compute the correlation map $\boldsymbol{\rho}$ for the image. A pixel q_i is deemed as tampered if the correlation ρ_i is smaller than a threshold t. In [23], the threshold is determined by the correlation distribution under hypothesis H_0 from Equation (2.5) by setting a constant false acceptance rate (CFAR) to 10^{-5}

$$\int_t^{\infty} p(x|H_0)\mathrm{d}x = 10^{-5}. \tag{2.17}$$

Compared to [21], [23] considers the false positives may be introduced as some pristine blocks could have small correlations due to saturated pixels or highly textured content. Thus, to address this problem, pixel q_i will be corrected as pristine if the following relationship regarding the predicted correlation distribution is satisfied:

$$\int_{-\infty}^{t} p(x|H_1)\mathrm{d}x > \beta, \tag{2.18}$$

, where β can be considered as the expected maximum of the false positive rate and is set to 0.01 in [23]. Due to the morphology of the tampered regions (for example, in an image, it requires a relatively large collection of pixels together to alter the image content), the resultant binary map will be further dilated with a square kernel of size 20×20 pixels to remove small detected regions to obtain the final forgery detection result.

2.1.4 Bayesian-MRF Based Method

The constant false acceptance rate (CFAR) method makes independent decision for each pixel which ignores the morphological meaning of the tampered region. Image forgeries are usually used by adding or hiding objects to and from an image. Thus, the tampered pixels often appear in clusters in the form of the objects. Without considering this behaviour, the detection result from the CFAR method made based on the independent pixel statistics could generate fragmented and inconsistent binary detection map. Thus, the underlying

spatial relationship between the pixels could be exploited for forgery detection. Chierchia *et al.* exploits this relationship by using a Bayesian Markov random field (MRF) based method in [24]. The Bayesian MRF-based method considers both the PRNU correlation and neighbouring pixels' relationships. The forgery detection is formulated as a binary labelling problem. For an image of size $M \times N$ pixels, the problem is to find the binary labelling map $\hat{\boldsymbol{u}} \in \{0,1\}^{M\times N}$ which maximises the probability of the occurrence given the correlation map $\boldsymbol{\rho}$:

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u}\in\{0,1\}^{M\times N}}{\operatorname{argmax}} \; p(\boldsymbol{\rho}|\boldsymbol{u}, \hat{\boldsymbol{\rho}})p(\boldsymbol{u}), \tag{2.19}$$

where $p(\boldsymbol{\rho}|\boldsymbol{u}, \hat{\boldsymbol{\rho}})$ is the conditional likelihood of observing the real correlation map $\boldsymbol{\rho}$ under the prior of having the predicted correlation map $\hat{\boldsymbol{\rho}}$ and the binary labelling map $\boldsymbol{u}$. This conditional likelihood can be estimated based on the correlation predictor from [23]. The probability $p(\boldsymbol{u})$ considers the spatial dependencies of the pixels by resorting to the MRF using Gibbs probability law:

$$p(\boldsymbol{u}) = \frac{1}{Z} e^{-\sum_{c\in\mathcal{C}} V_c(\boldsymbol{u})}, \tag{2.20}$$

where Z is a normalizing constant and $V_c(\cdot)$ is the potential. Modelling the potential $V_c(\cdot)$ using the Ising model [80, 81], the potential energy only takes the single-site cliques $\{c'\}$ and 4-connected two site cliques $\{c''\}$ into considerations and is the sum of the following two terms:

$$V_{c'}(u_i) = \begin{cases} -\frac{\alpha}{2}, & \text{if } u_i = 0, \\ \frac{\alpha}{2}, & \text{if } u_i = 1 \end{cases} \tag{2.21}$$

$$V_{c''}(u_i, u_j) = \begin{cases} \beta, & \text{if } u_i \neq u_j \\ 0, & \text{otherwise} \end{cases} \tag{2.22}$$

where the single-site potentials are directly related to the prior probability of being tampered p_0 and non-tampered p_1 with $\alpha = \log(p_0/p_1)$. β is the edge-penalty parameter, penalizing the adjacent pixels for having inconsistent labels. With $p(\boldsymbol{u})$ defined, Equation (2.19) can be rewritten as:

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u}\in\{0,1\}^{M\times N}}{\operatorname{argmin}} \Big\{ - \sum_{i=1}^{M\times N} \log p(\rho_i|u_i, \hat{\rho}_i) + \alpha \sum_{i=1}^{M\times N} u_i + \beta R(\boldsymbol{u}) \Big\}, \tag{2.23}$$

where the regularization term $R(\boldsymbol{u})$ is the sum of all class transitions over all four-connected cliques of the image:

$$R(\boldsymbol{u}) = \sum_{i=1}^{M\times N} \sum_{j\in\mathcal{N}_i} |u_j - u_i|, \tag{2.24}$$

with $\mathcal{N}_i$ the set of four-connected neighbours of pixel i.

By assuming the likelihood probability to be Gaussian under both hypotheses H_0 and H_1 from Equation (2.5), with zero mean and variance σ_0^2 under hypothesis H_0, and mean $\hat{\rho}_i$ and variance σ_1^2 under hypothesis H_1, Equation (2.23) becomes:

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u}\in\{0,1\}^{M\times N}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{M\times N} u_i \Big[\frac{(\rho_i - \hat{\rho}_i)^2}{2\sigma_1^2} - \frac{\rho_i^2}{2\sigma_0^2} - \log\frac{\sigma_0}{\sigma_1} - \log\frac{p_1}{p_0} \Big] + \beta R(\boldsymbol{u}) \Big\} \quad (2.25)$$

By resorting to the convex-optimization algorithm proposed in [82], the minimization problem can be solved and gives the optimal $\hat{\boldsymbol{u}}$. Compared to CFAR method from [23], this method considers the spatial relationships between the pixels and produces more consistent binary detection results.

2.1.5 Multi-scale Analysis Strategy Based Method

Both the methods proposed in [23, 24] use the block-wise correlation as the decision statistics for pixel-wise forgery detection. When the detection sliding window moves across an image with tampered regions, at certain point, the block covered by the window may include both pristine and tampered pixels. Thus, for such a heterogeneous block, neither the hypothesis H_0 nor H_1 from Equation (2.5) would stand for the whole block as some pixels have the camera's PRNU R and the others not. This would result in some tampered pixels or small forgeries not being correctly identified by the forgery detection methods. To address this problem, image segmentation based methods [83–85] were proposed in order to separate the regions. However, image segmentation itself is a sophisticated computer vision task and a good segmentation depends on several factors which might not be met, for example, the image might be noisy, highly textured or under-exposed. Furthermore, the image segmentation based methods assume the tampered region sharing the same boundary information as the image contents. However, this is not always true. For example, when an object is completely covered by the background as shown in Figure 2.2, the image segmentation based method will not be effective in segmenting the tampered and pristine regions. Hence, image segmentation based method is not the best solution for this problem.

Using a smaller detection window would naturally lower the occurrence of the heterogeneous blocks. However, due to the weak nature of PRNU, a smaller window size also means a large variance in the PRNU correlation, which will make the tampered and pristine correlations harder to be discriminated. Thus, there is a trade-off between using the smaller window sizes for more precise localization of tampered pixels and using the larger window sizes for better discriminability of the detection statistics. To balance this trade-off, Korus and

Original Image | Forged Image | Forgery Mask

Figure 2.2: An example of an image with tampered region covered by the background. Image segmentation based methods will not be able to estimate the tampered region using segmentation method.

Huang proposes the multi-scale analysis strategy in [86] to fuse the detection results using different detection window sizes.

Korus and Huang first consider the multi-scale analysis of the forgery detection as a fusion problem. For a set of sliding windows with increasing size $\{\omega_s\}$ for $s \in \{1, \ldots, S\}$, the goal is to fuse their resulting candidate maps generated using a single-scale forgery detector, like the aforementioned CFAR and Bayesian-MRF methods, to obtain an optimal binary detection map $\boldsymbol{t} \in \{0,1\}^{M \times N}$ for an image of size $M \times N$:

$$\left(\{\boldsymbol{c}^{(s)}, \{\boldsymbol{p}^{(s)}, \boldsymbol{y}\}\right) \rightarrow \boldsymbol{t} \in \{0,1\}^{M \times N}, \tag{2.26}$$

where $\boldsymbol{c}^{(s)} \in [0,1]^{M \times N}$ denotes the sth input candidate map corresponding to analysis window of size ω_s. Each candidate map has a corresponding reliability map $\boldsymbol{p}^{(s)} \in [0,1]^{M \times N}$, which identifies unreliable detection region, caused by factors like saturated pixels and highly textured contents. $\boldsymbol{y}$ represents the image content which is also used to guide tampering localization.

The fusion problem is formulated in terms of random fields and resolves to finding the optimal labelling of $\boldsymbol{t}$ (with $t_i = 1$ corresponding to tampered pixel at i) that minimises the following energy function:

$$\frac{1}{S} \sum_{i=1}^{M \times N} \sum_{s=1}^{S} E_\tau(c_i^{(s)}, t_i) + \alpha \sum_{i=1}^{M \times N} t_i + \sum_{i=1}^{M \times N} \sum_{j \in \mathcal{N}_i} \beta_{ij} |t_i - t_j| \tag{2.27}$$

The first term penalizes differences with respect to S candidates maps. The second term is a penalty term which can introduce a bias towards the hypothesis H_0 from Equation (2.5) with α being the weight. The third term penalizes inhomogeneity in a pixel's 8-connected neighbourhood, with $\mathcal{N}_i$ denoting the 8-connected neighbourhood of pixel i.

The potential $E_\tau(c,t)$ from the first term is defined as:

$$E_\tau(c,t) = -\log\max(\Psi_{\min}, \Psi_\tau(c,t)), \tag{2.28}$$

with $\Psi_{\min} \in [0,1]$ and:

$$\Psi_\tau(c,t) = \begin{cases} 1 - \frac{c}{2\tau} & \text{for } t = 0, \\ 1 + \frac{c}{2(1-\tau)} - \frac{1}{2(1-\tau)} & \text{for } t = 1, \end{cases} \tag{2.29}$$

where $\tau \in (0,1)$ is a quasi-threshold that equalises potentials for both decisions, i.e., $E_\tau(\tau,0) = E_\tau(\tau,1)$. $\Psi_{\min}$ is set to 10^{-3} in [86]. Drift thresholding [87] is applied to sliding windows of different scales:

$$\begin{cases} \tau^{(1)} & \text{if } s = 1, \\ \tau_i^{(s-1)} + \delta p_i^{(s-1)} & \text{if } s > 1 \text{ and } c_i^{(s-1)} \leq \tau_i^{(s-1)}, \\ \tau_i^{(s-1)} - \delta p_i^{(s-1)} & \text{if } s > 1 \text{ and } c_i^{(s-1)} > \tau_i^{(s-1)}, \end{cases} \tag{2.30}$$

where $\delta \in [0,1]$ is the strength of the drift and $\tau^{(1)}$ is an initial threshold. The drift is weighted proportionally to the region's reliability $\boldsymbol{p}^{(s)}$.

The reliability map $\boldsymbol{p}^{(s)}$ indicates regions with reliable detection with $p_i = 1$ while using $p_i = 0$ for unreliable detection. The inclusion of the reliability map in the first term can reset the score to eliminate false positive detections and facilitate easier score propagation through neighbourhood interactions. The reliability map is defined as:

$$p_i = 1 - e^{-\xi_0 |c_i - \frac{1}{2}|^{\xi_1}}, \tag{2.31}$$

where ξ_0 and ξ_1 are set to 30 and 2.5 empirically in [86].

β_{ij} denotes the weight for the neighbourhood interaction term from Equation (2.27):

$$\beta_{ij} = \beta_0 + \beta_1 e^{-\frac{1}{2}\phi^{-2}\|y_i', y_j'\|_{L_2}^2}, \tag{2.32}$$

where $\|y_i', y_j'\|_{L_2}$ denotes L_2 distance between two pixels in RGB color space. With this term, similar neighbouring pixels are more likely to have the same detection result.

With all the terms defined, the multi-scale fusion problem can be solved by minimizing the potentials in Equation (2.27). Apart from the multi-scale fusion based method, two other adaptive detection methods are also presented in [86], with one using segmentation-guided approach to decide the PRNU correlation detection window adaptively and the other dynamically choosing the optimal detection window size until a confident tampering probability estimation could be made. Despite the multi-scale and the adaptive methods

from [86] outperform the single-scale forgery detection methods, their base is using single-scale detection results according to the PRNU correlation statistics from the correlation predictor proposed by [23]. Thus, a correlation predictor with good precision under different scenarios becomes the foundation for generating reliable forgery detection results.

2.2 Poissonian-Gaussian Image Sensor Noise Modelling

2.2.1 Image Sensor Noise Modelling

In the previous subsection, we discussed different PRNU-based forgery detection algorithms and the important role of the correlation predictor from [23] plays. However, we found that the performance of the correlation predictor is not always optimal with several limitations. To understand the sources of these limitations, we have to revise the noise model the correlation predictor is built upon.

The noise model from Equation (2.6) is rather simplified. Despite it considers the camera gain and the gamma correction, the PRNU irrelevant noise in this model is simply formulated as an independent variable with no mentioning to the input signal. This is not accurate as the noise contains signal-dependent components. Without considering this signal-dependency, the conclusion derived from this simplified model could miss important factors which can impact the correlation predictor's performance.

To have a better understanding of this matter, a more detailed noise model for digital image sensor is required. An important work in this field is done in [88]. [88] considers a signal-dependent noise model by formulating both the Poissonian and Gaussian parts for noise in raw image data. [88] starts from a generic signal-dependent noise model:

$$z(x) = y(x) + \sigma(y(x))\xi(x), \tag{2.33}$$

where $x \in X$ is the pixel position in the domain X. z is the observed signal and y is the original signal. ξ is zero-mean independent random noise with the standard deviation equal to 1 and σ is a signal-dependent noise which gives the standard deviation of the overall noise component. By considering two mutually independent parts, a Poissonian signal-dependent component η_{p} and a Gaussian signal-independent component η_{g}, the noise $\sigma(y(x))\xi(x)$ can be written as:

$$\sigma(y(x))\xi(x) = \eta_{\mathrm{p}}(y(x)) + \eta_{\mathrm{g}}(x) \tag{2.34}$$

The two components are characterised as follows:

$$\chi(y(x) + \eta_{\mathrm{p}}(y(x))) \sim \mathcal{P}(\chi y(x)) \tag{2.35}$$

$$\eta_{\mathrm{g}}(x) \sim \mathcal{N}(0, b), \tag{2.36}$$

where χ and b are real scalar parameters and $\mathcal{P}$ and $\mathcal{N}$ denote the Poisson and Gaussian distribution, respectively. With the elementary properties of the Poisson distribution, the mean and the variance for the Poissonian component have the following relationship:

$$E\{\chi(y(x) + \eta_{\mathrm{p}}(y(x)))\} = \mathrm{var}\{\chi(y(x) + \eta_{\mathrm{p}}(y(x)))\} = \chi y(x). \tag{2.37}$$

Thus, the Poissonian component η_{p} has varying variance that is linearly proportional to $y(x)$ and the Gaussian component has a fixed variance b. The overall variance of the noise model conforms to:

$$\sigma^2(y(x)) = ay(x) + b, \tag{2.38}$$

with a being the linear coefficient.

Considering the physical meaning of each component, we will see how the above Poissonian-Gaussian model is naturally suited for the raw-data of digital image sensors. The Poissonian component η_{p} models the photon-counting process at each pixel on a sensor. The Gaussian component η_{g} accounts for the signal-independent errors such as electric and thermal noise. The scalar parameter χ is related to the quantum efficiency of the sensor, which measures the sensor's photoelectric conversion rate, which determines the amount of electric charge the sensor collects. But in addition to the above model, in digital image sensors, the collected charge is always added to some base "pedestal" level p_0. This constitutes an offset-from-zero of the output data and it can be rewritten as a shift in the argument of the signal-dependent part of the noise:

$$\begin{aligned} \mathring{z}(x) &= \mathring{y}(x) + \mathring{\sigma}(\mathring{y}(x) - p_0)\mathring{\xi}(x) \\ &= \mathring{y}(x) + \mathring{\eta}_{\mathrm{p}}(\mathring{y}(x) - p_0) + \mathring{\eta}_{\mathrm{g}}(x) \end{aligned} \tag{2.39}$$

Notice the above expression is denoted by the circle superscript $\mathring{}$. It is done to differentiate the expression from the final output, which needs to be amplified by the analogue gain. The analogue gain is the amplification of the collected charge. The amplification Θ is formalised as the multiplication of the noise-free signal, of the Poissonian noise, and of a part of the Gaussian noise by a scaling constant θ:

$$z(x) = \Theta(\mathring{z}(x)) = \theta(\mathring{y}(x) + \mathring{\eta}_{\mathrm{p}}(\mathring{y}(x) - p_0) + \mathring{\eta}'_{\mathrm{g}}(x)) + \mathring{\eta}''_{g}(x). \tag{2.40}$$

In the above equation, the Gaussian noise term $\mathring{\eta}_{\mathrm{g}}$ has been split in two components $\mathring{\eta}'_{\mathrm{g}}$ and $\mathring{\eta}''_{\mathrm{g}}$, where $\mathring{\eta}''_{g}$ represents the portion of the noise that is introduced after the amplification and thus not affected by the factor θ. The expectation and variance for z are:

$$E\{(x)\} = y(x) = \theta\mathring{y}(x) \tag{2.41}$$

$$\mathrm{var}\{z(x)\} = \theta^2\chi^{-1}(\mathring{y}(x) - p_0) + \theta^2\mathrm{var}\{\mathring{\eta}'_{g}(x)\} + \mathrm{var}\{\mathring{\eta}''_{\mathrm{g}}(x)\}. \tag{2.42}$$

Hence, the above expression conforms to the form of Equation (2.38) with:

$$\begin{cases} a = \chi^{-1}\theta \\ b = \theta^2\mathrm{var}\{\mathring{\eta}'_{g}(x)\} + \mathrm{var}\{\mathring{\eta}''_{g}(x)\} - \theta^2\chi^{-1}p_0 \end{cases} \tag{2.43}$$

In digital cameras, the analogue gain θ is controlled by the choice of ISO speed settings. Thus, this Poissonian-Gaussian model shows how the variance of the noise is dependent on the ISO speed. However, so far, the model has not taken the PRNU into consideration. As PRNU is the non-uniform response of the pixels to the light, which originates from the slight difference of the quantum efficiencies at pixels, so it will influence the terms with χ. Also, from observation, we found that the terms with χ are also correlated with the amplification factor θ, indicating the ISO speed will have an impact on the noise components related to the PRNU in the raw data. This relationship needs to be further investigated to evaluate how the ISO speed would impact the PRNU correlation and detailed studies are presented in Chapter 4.

2.2.2 Local Estimation of the Expectation and Variance for the Noise Model

Foi *et al.* [88] not only formulate the Poissonian-Gaussian noise model, but also provide a method to estimate the sensor signal y_i and its variance $\sigma^2(y_i)$ from noisy raw images. This method could be useful for the validation of the PRNU noise model. [88] facilitates the noise analysis through wavelet domain analysis. It considers the wavelet detail coefficients z^{wdet}, which is defined as the downsampled convolution:

$$z^{\mathrm{wdet}} = \downarrow_2 (z \circledast \psi), \tag{2.44}$$

where $\downarrow_2$ denotes the downsampling operation and ψ is a 2-D wavelet function with zero mean and unity l^2-norm. Similarly, the normalized approximation coefficients, z^{wapp}, are defined as:

$$z^{\mathrm{wapp}} = \downarrow_2 (z \circledast \varphi), \tag{2.45}$$

where φ is the corresponding 2-D wavelet scaling function, which is normalized so that $\sum \varphi = 1$.

For noisy images, the detail coefficients z^{wdet} contain mostly noise and it gives

$$\begin{aligned} \text{std}\{z^{\text{wdet}}\} =& \downarrow_2 (\text{std}\{z \circledast \psi\}) = \downarrow_2 (\sqrt{\text{var}\{z\} \circledast \psi^2}) \\ \approx& \downarrow_2 (\text{std}\{z\}\|\psi\|_2) = \downarrow_2 (\text{std}\{z\}) \\ =& \downarrow_2 (\sigma(y)) = \sigma(\downarrow_2 y) = \sigma(\downarrow_2 (y \sum \varphi))' \\ \approx& \sigma(\downarrow_2 (y \circledast \varphi)) = \sigma(E\{z^{\text{wapp}}\}) \end{aligned} \tag{2.46}$$

with the approximation becoming accurate in regions, where y is uniform. Thus, for pixel x in uniform regions, it can be assumed that

$$z^{\text{wdet}}(x) \sim \mathcal{N}(0, \sigma(E\{z^{\text{wapp}}(x)\})) \tag{2.47}$$

As $\sum \varphi = 1$, it is always true that $\|\varphi\|_2 \neq 1$. Therefore, when considering $\text{std}\{z^{\text{wapp}}\}$, Equation (2.46) comes to:

$$\text{std}\{z^{\text{wapp}}\} \approx \|\varphi\|_2 \sigma(z^{\text{wapp}}) \tag{2.48}$$

[88] uses separable kernels with $\psi = \psi_1 \circledast \psi_1^T$ and $\varphi = \varphi_1 \circledast \varphi_1^T$, where ψ_1 and φ_1 are 1-D Daubechies wavelet and scaling functions.

To better facilitate the uniform region assumption, [88] segments an image into level sets, in each of which the image can be reasonably assumed to be uniformly close a certain value. Spatial smoothing is deployed to attenuate the noise in an image to better estimate the segmentations. Also, an edge detector is used to avoid the impact from edges in the noise analysis.

The spatial smoothing is done by convolving the normalized approximation coefficients with a uniform 7×7 kernel $\bar{\omega}$:

$$z^{\text{smo}} = z^{\text{wapp}} \circledast \bar{\omega}, \tag{2.49}$$

where $\|\bar{\omega}\|_1 = 1$. After this smoothing operation, in the corresponding regions where y itself is smooth, z^{smo} is approximately equal to $E\{z^{\text{wapp}}\}$ and thus to $\downarrow_2 y$.

The edge detection is carried out by setting a threshold for the smoothed derivatives of the image against an estimate of the local standard deviation. As the mean of the absolute deviations of $\mathcal{N}(0,1)$ is equal to $\sqrt{2/\pi}$, a map for the rough estimation of the local standard deviations of z^{wdet} can be defined as:

$$s = \sqrt{\frac{\pi}{2}} |z^{\text{wdet}}| \circledast \bar{\omega}. \tag{2.50}$$

Base on this map, the set of smoothness X^{smo} is defined as:

$$X^{\text{smo}} = \{x \in\downarrow_2 X : |\nabla(\Lambda(z^{\text{wapp}}))(x)| + |\Lambda(z^{\text{wapp}})(x) < \tau \cdot s(x)\}, \tag{2.51}$$

with

$$\Lambda(z^{\text{wapp}}) = \nabla^2 \text{medfilt}(z^{\text{wapp}}), \tag{2.52}$$

where ∇ and ∇^2 are gradient and Laplacian operators, respectively. medfilt is a 3×3 median filter. $\downarrow_2 X$ denotes the decimated domain of the wavelet coefficients z^{wapp} and τ is a positive threshold. By thresholding the sum of the gradient and the Laplacian, it provides a heuristic way to obtain thickened edges.

With the smoothed images and edges excluded, the pixels can be divided into level sets to better facilitate the uniform region assumption. [88] divides the smoothness set X^{smo} into a collection of N non-overlapping level sets $S_i \subset X^{\text{smo}}, i = 1, \ldots, N$. For each level set, it is characterised by its center value u_i and allowed deviation $\Delta_i > 0$, which is defined as:

$$S_i = \{x \in X^{\text{smo}} : z^{\text{smo}}(x) \in [u_i - \Delta_i/2, u_i + \Delta_i/2)]. \tag{2.53}$$

As for the pixels in the smoothness set, $x \in X^{\text{smo}}$, they have the following properties:

$$z^{\text{smo}}(x) = E\{z^{\text{wapp}}(x)\} = E\{(\downarrow_2 z)(x)\} = (\downarrow_2 y)(x) \tag{2.54}$$

$$\text{std}\{z^{\text{wdet}}(x)\} = \text{std}\{(\downarrow_2 z)(x)\} = (\downarrow_2 (\sigma(y)))(x). \tag{2.55}$$

Thus, for each level set S_i, the local estimation of pixel's expectation value can be estimated as $\hat{y}_i$:

$$\hat{y}_i = \frac{1}{n} \sum_{j=1}^{n_i} z^{\text{wapp}}(x_j), \quad \{x_j\}_{j=1}^{n_i} = S_i. \tag{2.56}$$

The standard deviation of $\sigma(y_i)$ can be calculated as the unbiased sample standard-deviation of the detail coefficients z^{wdet} on S_i:

$$\hat{\sigma}_i = \frac{1}{\kappa_{n_i}} \sqrt{\frac{\sum_{j=1}^{n_i} (z^{\text{wdet}}(x_j) - \bar{z}_i^{\text{wdet}})^2}{n_i - 1}}, \tag{2.57}$$

where $\bar{z}_i^{\text{wdet}} = \frac{1}{n_i} \sum_{j=1}^{n_i} z^{\text{wdet}}(x_j)$ and the factor $\kappa_{n_i}^{-1}$ is defined as:

$$\kappa_n = \sqrt{\frac{2}{n-1}} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})} = 1 - \frac{1}{4n} - \frac{7}{32n^2} + O(\frac{1}{n^3}), \tag{2.58}$$

where $O(\frac{1}{n^3})$ omits the higher order errors. The factor κ_n comes from the mean

of the chi-distribution with $n-1$ degrees of freedom.

Equation (2.56) and (2.57) provide a way to estimate the expected pixel values and pixels' standard-deviations, hence the variances as well, from noisy raw images. This method will be used in Chapter 4 to validate the sensor noise model involving the PRNU term developed in our work.

2.3 PRNU-based Source-oriented Image Clustering

Both PRNU-based image forgery detection discussed previously and image source camera identification use reference PRNU from candidate cameras to determine an image's originality and integrity. The reference PRNU has to be extracted from a set of images taken by the same camera. This requirement could be fulfilled if the source camera is available but the availability of the source camera cannot be assumed in many real-world scenarios. For example, the forensic investigators may need to carry out investigations on social network accounts, checking whether and which accounts share images from the same device and thus, discover the underlying connection between these accounts. Under such a scenario, the forensic investigators may not have access to the source cameras. All the information available is the set of images from these accounts. With the goal of the provenance analysis for these images is to group them into clusters based on their source devices, the forensic investigators have to exploit the relationship between images. The pairwise correlations between the PRNUs extracted from the images could provide vital source information. However, again, due to the weak nature of the PRNU, the correlation between two PRNUs extracted from two single images could contain strong noise. To take this factor into account, a clustering algorithm needs special designs to be robust to this noise. Thus, several methods have been proposed for PRNU-based source oriented image clustering.

2.3.1 Markov Random Field Based Methods

One of the first works in source oriented image clustering is proposed in [89]. [89] treats each PRNU extracted from each image as a random variable and uses the Markov random field approach to iteratively assign the PRNUs into clusters. Firstly, a subset of M images are randomly selected from all the images to form a training set. Each image forms a singleton cluster and the pairwise correlations are computed between the M clusters. Thus, each cluster's corresponding PRNU has $M-1$ correlations computed with the others. With the correlations between the clusters computed, for each cluster, k-means clustering with $k=2$ is applied to the $M-1$ correlations to group the correlations into two groups (one as intra-class and the other as inter-

class). The average of the centroids for the two clusters is used as a reference correlation. Also, a membership committee consisting of a certain number of the most similar PRNUs from the training set is formed for each PRNU in the training set. The reference correlation and the membership committee are used to estimate the likelihood probability of assigning each class label to the corresponding PRNU. The class label with the highest probability in its membership committee is assigned to the PRNUs in question. The clustering steps can be performed iteratively and stops when there are no label changes after two consecutive iterations. With the clusters formed in the training data, each PRNU from the rest of the dataset is classified to its closet cluster to form the final result. Being one of the earliest works in source oriented clustering, while this method performs well on small dataset without the need for the priori knowledge about the dataset, this method has several limitations. First, the likelihood computation complexity is $\mathcal{O}(n^3)$ with respect to the number of images, which makes the method run extremely slow on large dataset. Secondly, as the final clusters are formed by using the clusters from the training data to attract the rest of the images, to ensure there is a correct cluster for each image in the training set, the number of image selected for training set, M, needs to be large. A large M will make the likelihood probability computation more expensive.

A faster MRF based method is presented in [25]. Similar to [89], the algorithm starts from singleton clusters and iteratively updates each image's label with the usage of a membership committee for each cluster. But some key differences make the performance of this method superior to the one from [89]. Instead of using correlation as the similarity measurement, [25] uses the shared nearest neighbours (SNN). SNN-based clustering algorithms have shown good performance in finding clusters of different sizes and densities. Thus, the usage of SNN in [25] could help in this aspect. As the likelihood probability computation in [89] is very time-consuming, a concise yet effective cost function is used in [25] to address this problem. The cost function enables the clustering results to converge accurately and efficiently. Furthermore, [25] uses deterministic relaxation for the MRF to update the labels for images, which accelerates the rate of convergence. Overall, this method can group images into clusters very efficiently. In contrast to the method from [89] which splits the clustering task into two stages by constructing a training set, this method can be applied to a large dataset directly without worrying about not finding a representative cluster for each image during the training stage.

2.3.2 Hierarchical Clustering Based Methods

A hierarchical clustering based method is proposed in [90]. Similar to [89], only a subset of the images is used as the training set for the clustering stage and the rest of the images are attracted to the cluster centroids as a classification problem. The method starts from using singleton clusters in the training set as well. The algorithm calculates the pairwise similarity matrix for the training set and merges clusters iteratively. For each iteration, two closest clusters are merged and the correlation matrix is updated with the correlations between the newly formed cluster and all other clusters. After each iteration, a silhouette coefficient, which measures the separation among clusters and the cohesion within each cluster, is calculated for the cluster. A global measure of the silhouette coefficients is recorded by taking the average over all the clusters. This algorithms runs iteratively until all the images are merged into a single cluster. At the end of the iterative process, an optimal partition will be decided for the one with the lowest global silhouette coefficient. After finishing the clustering of the images from the training set, the rest of the dateset is classified into the formed clusters. A similar hierarchical clustering method is presented in [91]. Despite that the hierarchical clustering is reportedly faster than the method presented in [89], the iterative merging process is still computationally expensive. Also, the accuracy of the method is also dependent on the size of the training set.

2.3.3 Graph Clustering Based Methods

[92] treats the image clustering as a graph partition problem using a weighted undirected graph. Each image's PRNU is considered as a vertex in the graph and the weight of each edge is represented by the similarity between the two vertices linked by the edge. To avoid the time-consuming pairwise similarity computation, [92] constructs a sparse graph instead. A vertex is randomly selected as the initial center of the graph and the weights of its edge to all other vertices are then calculated. The $(\kappa + 1)th$ closest vertices to the initial center is selected as the second centers. The weights of edges from these vertices to all other vertices are calculated as well. κ is a parameter, which controls the sparsity of the graph. The construction procedure stops when the number of vertices not considered as centers is less than κ. After the construction procedures stops, the multi-class spectral clustering (MCSC) [93] is applied to the constructed graph to partition the vertices into a number of clusters. However, a drawback for this method is about the stopping criterion for the iterative partition. To find the optimal partition, the algorithm works in an iterative manner until the size of the smallest cluster equals 1. Thus, to overcome this drawback, [94] proposes the use of the silhouette coefficient

to find the optimal partition. Nonetheless, despite the usage of the silhouette coefficient, the randomness of the MCSC still exists.

To address the randomness in the performance caused by the MCSC, [95] proposes a normalized cuts (NC) [96] based clustering method. Again, the method considers the clustering as a graph partition problem with the PRNUs as the vertices and the pairwise similarities between the PRNUs as the weights for edges connecting the vertices. A cut means the partition of a graph into two disjoint graphs by the removal of the edges connecting the two parts. The total weight of the removed edges gives a computation of the degree of dissimilarity between the two parts, defined as the cut value. The optimal bipartition of a graph is obtained by means of the minimization of the cut value. However, this would favour cutting small sets of isolated nodes in the graph. Thus, to avoid such a behaviour, instead of using the cut value directly, a normalized cut is used and defined as sum of the cut values divided by the total connections from each partition to all the nodes in the original graph. The clustering can be carried out by minimizing the normalized cut and bipartitioning the graph iteratively. Though the normalized cut method addresses the randomness from MCSC and yields more stable performance, the choice of the stopping criterion for this method is critical to the performance. The stopping criterion is based on the comparison of an aggregation coefficient, defined as the mean value of the edge weights inside a cluster, with a pre-defined threshold. In [96], the optimal threshold value is estimated by preliminary experiments on a training set. However, in real-life scenario, it might be difficult to find training images sharing the exact statistical characteristics as the image in questions.

2.3.4 Consensus Correlation Clustering Based Method

With all the aforementioned methods requiring users to either set the size of the training set or use priori knowledge to find the optimal threshold, a consensus correlation clustering based method is proposed in [97], which does not require any user-defined parameters. Correlation clustering is essentially a graph-based clustering algorithm. [97] runs the correlation clustering using a fully-connected graph. It formulates the graph partition problem as a constrained energy minimization problem with the energy defined as the sum of the weights for all cut edges. In [97], the weights for the edges are measured as the similarity between the connected vertices but with a constant shift. As the goal is to minimize the energy, the constant shift will exert a tendency shift for the algorithm. With a large and positive shift, the method tends to make less cuts and therefore, forms a single cluster. A negative shift will encourage the method to make more cuts and more likely to form singleton clusters. Thus, the correlation clustering itself needs to find an optimal setting for this shift

constant. Instead of finding an optimal setting for the correlation clustering, [97] runs correlation clustering with 50 different settings of the constant shift. With an ensemble of clustering results formed, consensus clustering is applied. The large ensemble of the correlation clustering results are fed to the Weighted Evidence Accumulation Clustering (WEAC) to obtain the consensus clustering result. The consensus clustering result is further refined by using log-likelihood estimators to check whether some clusters can be merged with one of the large clusters, by utilizing the improved PRNU estimation due to the large cluster size.

Finding an optimal and efficient way to conduct PRNU-based source oriented clustering is still an open question. Different application scenarios can put different constraints on the problem. Thus, many other works [5, 26, 98–101] have been conducted in this field as well. However, all the above mentioned clustering methods consider the similarity measurement between pristine images' PRNUs. With the fact that some common image editing tools may impact the similarity measurement, especially on social networking sites, their influence on the clustering performance needs to be investigated.

2.4 Anti-forensics Attacks on PRNU and Countering Methods

As shown in the previous sections, PRNU is a powerful image forensic tool, which can be used effectively for source camera, image clustering and forgery detection. Thus, to avoid being traced or detected by the PRNU-based forensic methods, the image attackers may target the PRNU to make those identification methods ineffective. Thus, these anti-forensics attacks pose direct threat to PRNU-based forensic methods. Correspondingly, forensic investigators would like to develop detection and countering methods on these attacks. Detecting PRNU attacked images can prevent PRNU-based methods from being applied to these images, which may lead the investigators to wrong conclusions. In addition, usually PRNU attacked images also have high probability with their contents being tampered as well. Thus, revealing these attacked images can also raise suspicion about the authenticity of their contents and thus, not to be deceived by them.

Generally speaking, attacks on PRNU can be categorised into two groups, (1) by introducing pixel-level misalignments and (2) suppression of PRNU. We will review both categories and the corresponding detection methods.

Figure 2.3: An example image showing how seams are found within an image. The red seams run vertically through the image, mostly running through the flat background which contains less information.

2.4.1 Attacks on PRNU by Disturbing Pixel Alignment

As the PRNU is a pixel-level signal, the computation of the similarity between PRNUs requires good alignment. Thus, introducing pixel-level misalignment is an effective way to disturb the methods which rely on the PRNU similarity. Simple geometric transformations, like resizing, rotation, cropping, *etc.*, are effective enough to introduce misalignment and anonymise images. However, it is shown that the parameters for these simple transformations could be determined by a brute-force search. With the transformation parameters determined, the PRNU can still be aligned with the distorted image [102, 103] and hence, the PRNU-based methods can remain effective. With this in mind, attackers may use an irreversible transformation whose parameters cannot be found using brute-force search to anonymise an image.

A seam-carving based method is proposed in [104] by Bayram *et al.* Seam-carving [105] is a content-aware image resizing method. A seam is defined as a connected path which runs either horizontally or vertically throughout an image with an example image shown in Figure 2.3 [1]. An image can be resized by removing a certain number of seams and then all remaining pixels are shifted horizontally or vertically to fill the gap. Each seam is obtained by measuring the gradient information at each pixel as an energy function and a path that minimises the energy is selected as the seam. Thus, a seam can be considered as a set of connected pixels which contains the least content information. Dirik *et al.* did an detailed analysis of seam-carving based anonymization attack on PRNU in [107]. They found that in order to achieve successful anonymization of the source camera for an image, there should not be many uncarved blocks larger than the size of 50×50 pixels. As the seam-carving is

[1] The image is excerpted from [106]

an irreversible manipulation which removes pixels from an image, to meet the requirement of not having many uncarved blocks will inevitably downgrade the image quality and may alter the image content as well. In addition, a source camera identification method for seam-carved images is proposed in [106], given the knowledge that multiple seam-carved images are from the same camera. The authors of [106] show that by combining the PRNUs from multiple seam-carved images with the same source, the source camera identification is still possible even if the sizes of the uncarved blocks in the images are less than the recommended size of 50×50 pixels.

2.4.2 Attacks on PRNU by Suppression of PRNU

With the PRNU being a high-frequency signal, it can be easily attenuated or even removed by some simple manipulation, including denoising and median filtering. Ever since the first PRNU-based method is proposed in [20], the authors have tested removing PRNU using denoising filters. It is found that the denoising filter used in [20] could decreases the correlation value between the extracted PRNU and the reference. By repetitive application of a denoising filter or use some aggressive denoising filters, the PRNU can be sufficiently suppressed to prevent successful source camera identification. Other low-pass filtering manipulations like median filtering, wavelet transform based low-pass filtering and Wiener filter are found to be effective in removing or suppressing the PRNUs in images [108, 109]. Thus, detecting these attacks have drawn attentions from the digital forensic research community.

Median Filtering Detection

Median filtering is a nonlinear operation which can preserve edges within an images. It is commonly used to perform image denoising, removing outlying pixel values and image smoothing. Thus, attackers may use median filtering to suppress an image's PRNU while preserving the edge information. Several early techniques had been developed to detect median filtering [50–53], but their detection performance could be downgraded in several important real-life scenarios. For example, they generally do not perform well on JPEG compressed images. To address this problem, Kang *et al.* propose a detection method [54] that analyses the statistical properties of an image's median filter residual defined as the difference between an image and its median filtered version. The median filter residual is fitted to an autoregressive (AR) model to extract the AR coefficients. A support vector machine (SVM) is used in [54] as a binary classifier to detect whether median filtering is applied. Using the extracted autoregressive coefficients as the features, the median filtering detection can be performed.

Wavelet-based Compression Detection

Not only the wavelet-based compression can suppress PRNU, it is also pointed out in [110] that wavelet-based method can be used to hide other image manipulation footprints. Thus, developing method to discriminate uncompressed and wavelet-based compressed images would contribute to the forensic community. In [110], anti-forensics attacks are carried out by dithering the discrete wavelet transform (DWT) histogram, which removes the quantization artifacts in the DWT histogram. In [111], Wang *et al.* notice that despite the dithering operation from [110] can successfully remove the quantization artifacts, the image spatial-domain information is not taken into account. The magnitudes of DWT coefficients at the same spatial location across different levels are highly correlated, especially for the typical localized image structured, such as edges. Thus, this inspired Wang *et al.* to study the statistical change in the image caused by the wavelet-based compression and dithering using a joint histogram of DWT coefficients across different levels. The Hough transformation is applied to the joint DWT histogram and the first four standardised moments of the Hough transform parameters are used as features. A SVM-based binary classifier can be trained by using the four features to detect whether an image is subject to wavelet-based anti-forensics attacks.

Generic Manipulation Detection

The methods developed in [54] and [110] construct specific features to detect specific types of anti-forensic attacks. With the vast number of the types of potential attacks on PRNU, constructing a specific set of features for every type of attacks becomes infeasible. Thus, researchers have been working in the direction of constructing universal feature sets that can be used to detect multiple types of attacks. Several works [112–115] were done in this field.

As PRNU is a high-frequency signal, the studies of the attacks on PRNU often focus on the high-frequency component in an image. For example, Verdoliva *et al.* extract features from the high-pass residual of an image in [113]. The residual image is extracted using a linear high-pass filter of the third order:

$$r_{ij} = x_{i,j-1} - 3x_{i,j} + 3x_{i,j+1} - x_{i,j+2}, \tag{2.59}$$

where x and r are the original and residual images, respectively. i and j are the Cartesian coordinates of a pixels.

The residual image is quantised such that a manageable number of bins can be set for the histogram of co-occurrences for the residual. The quantization and truncation is performed as:

$$\hat{r}_{ij} = \text{trunc}_{(T)}(\text{round}(r_{ij}/q)), \tag{2.60}$$

where T and q are the truncation value and the quantization step, respectively. They are set to $T = 2$ and $q = 1$ in [113]. The co-occurrence is computed on four pixels in a row:

$$C(k_0, k_1, k_2, k_3) = \sum_{i,j} \hat{r}(q_{i,j} = k_0, q_{i+1,j} = k_1, q_{i+2,j} = k_2, q_{i+3,j} = k_3). \quad (2.61)$$

This eventually gives a histogram $\boldsymbol{h}$ of 625 bins.

To use this residual-based feature for manipulation detection, Verdoliva *et al.* consider a binary hypothesis test, with H_0 being genuine image blocks and H_1 for tampered. As [113] considers H_1 being image blocks with heterogeneous sources, the detection problem is tackled through H_0. The H_0 training samples are fitted through a multidimensional Gaussian distribution. The mean vector and covariance matrix for the multidimensional Gaussian for the features $\boldsymbol{h}$ is defined as:

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{h}_n \\ \boldsymbol{\Sigma} &= \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{h}_n - \boldsymbol{\mu})(\boldsymbol{h}_n - \boldsymbol{\mu})^T \end{aligned} \quad (2.62)$$

With the distribution formulated, for each new feature under test $\boldsymbol{h}'$, the log-likelihood with respect to the Gaussian model can be computed as:

$$L(\boldsymbol{h}') = (\boldsymbol{h}' - \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{h}' - \boldsymbol{\mu})) \quad (2.63)$$

The log-likelihood can be compared with a threshold to make the final detection. Though this method is designed for heterogeneous manipulations, the choice of the threshold is challenging for this heterogeneous setting. In addition, this method is a local descriptor using a block-wise comparison method, which means the multidimensional Gaussian model built from one block can only be reliable for the image blocks from the same position of the same camera. Thus, it cannot be applied to images from other sources. But overall, this method shows the potential of using high-pass residuals to detect generic anti-forensics manipulations.

With the emergence of convolutional neural-network (CNN) based classifiers and their successful application in the closely related research fields of computer vision and pattern recognition, the digital forensic community starts to use neural networks for detecting generic anti-forensics attacks as well. Inspired by the usage of high-pass residual based features, Bayar and Stamm [77] proposed a manipulation detection method using a constrained CNN architecture with the first convolutional layer forced to perform high-pass filtering. In [78], Cozzolino *et al.* further prove that local features can be extracted using CNN

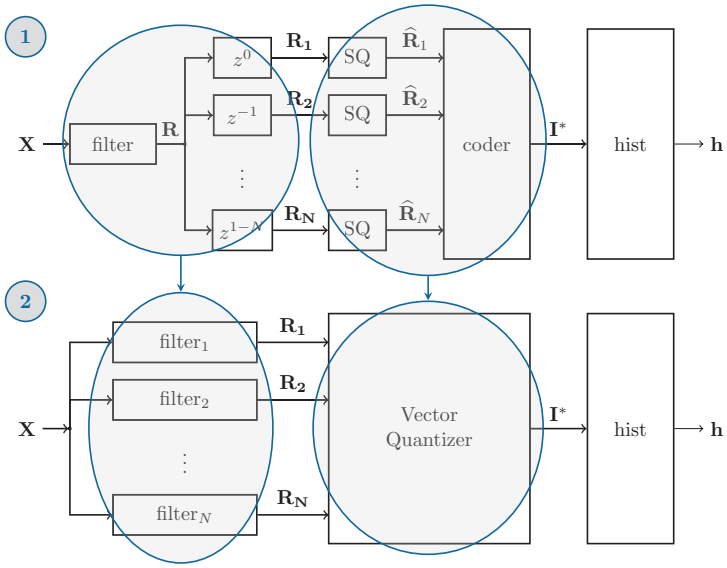


Figure 2.4: The processing Scheme (1) for extracting a spatial rich model with filter and shifters can be replaced by a bank of filters in Scheme (2). The bank of independent scalar quantization (SQ) and coder can be replaced by a vector quantiser. The figure is excerpted from [78].

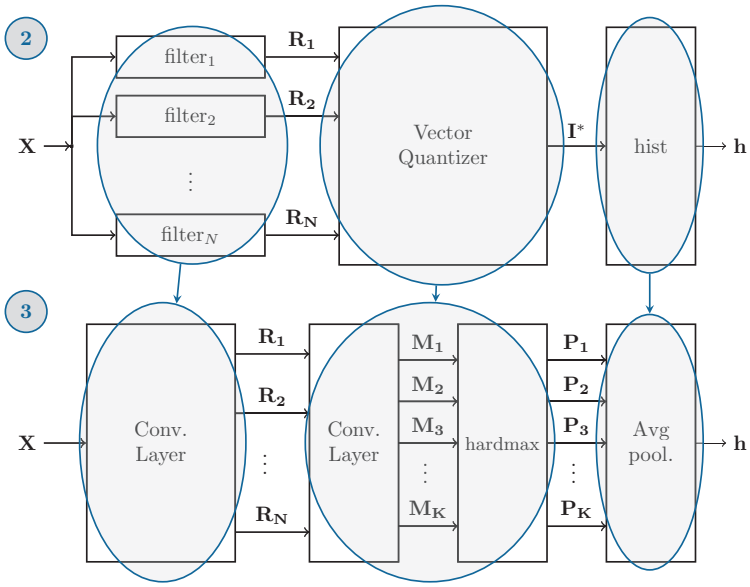


Figure 2.5: Scheme (2) can be converted to the CNN shown by Scheme (3). The bank of filters can be replaced by the convolutional layers and the vector quantiser can be replaced by convolutional hardmax layers. The histogram computation can be done through an average pooling layer. The figure is excerpted from [78].

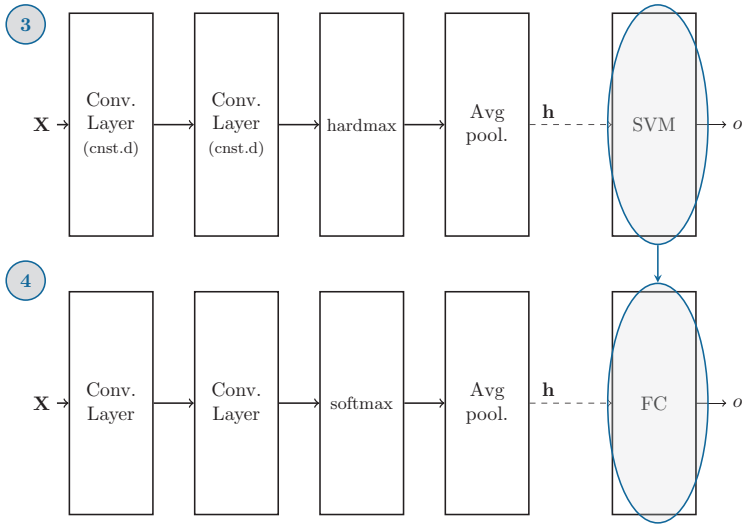


Figure 2.6: The constrained CNN shown in Scheme (3) with the extracted features fed to an external classifier (SVM) can be replaced by a fully connected layer with all constraints removed. The figure is excerpted from [78].

even without the high-pass filtering constraint. They prove it step-by-step, moving from local features to a Bag-of-Words (BoW) paradigm, and then proceed to the implementation of CNN as shown in Figure 2.4, 2.5 and 2.6 [2].

Scheme (1) in Figure 2.4 shows the feature extraction process for the spatial rich model, which is a model built on residual-based local descriptors, similar to the method from [113] reviewed above. X is the input image, which is passed through a filter to extract the residual image R. Afterwards, the residual image is shifted by pixels using z to create N versions of the residual images. The N shifted residual images are then quantised and encoded as I^*. The histogram of the I^* can be computed and used as feature, for example, like in Equation (2.62) and (2.63). Cozzolino *et al.* point out that the filter and the shifters from Scheme (1) can be replaced by a bank of filters, all identical to one another except for the position of the non-zero weights (for example, the weights in Equation (2.59), which are $[1, -3, 3, 1]$). The quantiser and coder group can be regarded as a constrained form of vector quantization. Thus, the structure and function of Scheme (1) is equivalent to Scheme (2) and Scheme (2) actually implements the Bag-of-Words paradigm.

From the BoW paradigm, Cozzolino *et al.* further point it out that the scheme can be implemented through a CNN. The bank of filters from Scheme (2) can be replaced by a convolutional layer and the vector quantiser can be replaced by convolutional-hardmax layers. The computation of the histogram

[2]The images are excerpted from [78]

can be done through an average pooling layer as shown in Figure 2.5 to extract the residual-based features. These features are then passed to an external classifier, for example, a SVM. Cozzolino *et al.* replace the external classifier with internal fully-connected layers to complete the CNN structure. Hence, they prove the residual-based local descriptors can be entirely extracted by a CNN.

Yu *et al.* show that the features extracted using a CNN can discriminate multiple types of anti-forensics manipulations [79]. However, despite the CNN can discriminate different types of anti-forensics manipulations, it requires the training set to include every type of possible manipulations. In real-life scenario, before the forensic investigators applying PRNU-based methods, often the question of whether an image's PRNU is attacked needs to be answered, no matter which type of manipulation it has undergone. For a training set with a finite size, it would not be able to include all potential types of manipulations. Thus, the binary classification of discriminating whether an image's PRNU is attacked or not remains an open question.

2.5 Summary

In this chapter, we first introduced different methods on PRNU-based image forgery detection. We show how the PRNU correlation predictor is developed and its important role in different forgery detection methods. We then revised a Poissonian-Gaussian sensor noise model, which gives us some hints about the ISO speed's impact on the noise component corresponding to the PRNU in an image. As this impact may put limitations on the existing correlation predictor, more detailed investigations are required. In addition, different PRNU-based source-oriented clustering methods are revised with detailed explanations on how they use the similarity measurements between the PRNUs to group images into clusters. These methods assume the similarity measurements are pristine. But with the emergence of common image editing tools used by social networks sites, the validity of this assumption is challenged and the impact from these editing tools needs to be investigated. We then discussed anti-forensics attacks on PRNUs and the corresponding detection methods. CNN based classifiers have been proved to be effective for detecting different types of manipulations but the binary classification of detecting whether an image's PRNU is attacked or not remains an open question. With the aforementioned problems and challenges in mind, studies are carried out in the following chapters to investigate and address these problems.

Chapter 3

Warwick Image Forensics Dataset

In Section 2.2 of Chapter 2, we discussed how the ISO speed may have an impact on the PRNU correlation and image forgery detection. Thus, studies need to be carried out to further investigate this phenomena and develop effective methods to improve the forgery detection performance on images of different ISO speeds. In order to carry out these studies, we need to have a large number of images taken at different ISO speeds. However, existing forensic datasets cannot meet this requirement on the quantity of the images taken at different ISO speeds. Therefore, it is necessary for us to construct a new dataset to facilitate the study of ISO speed's impact on PRNU-based image forensics.

In this chapter, we present a novel forensics image dataset, namely the Warwick Image Forensics Dataset. The dataset consists of more than $58,600$ images from 14 different cameras. The images are taken with special attentions to the camera exposure settings, especially the ISO speed and the exposure time, as well as using exposure bracketing and burst shot functions to take multiple frames of the same scene. With these special designs for the image compositions, not only the study of the ISO speed's impact on PRNU-based image forgery detection allowed to be carried out, but also it enables different multi-frame merging algorithms (e.g. HDR imaging) to be applied to the images such that forensic techniques for images subject to these algorithms can be developed as well. Thus, this dataset can provide a solid platform for studies on different topics in image forensics.

The rest of the chapter is organised as follows. In Section 3.1, a brief overview of the background, including existing forensic datasets, will be discussed. The details of the Warwick Image Forensics Dataset are presented in Section 3.2 and experimental evaluations are carried out in Section 3.3. A conclusion is given in Section 3.4.

3.1 Introduction

As introduced in the previous chapter, PRNU-based device fingerprinting methods have important roles in digital image forensics. Many studies have been carried out in this field. Public datasets like Dresden Image Dataset [116] and VISION Image Dataset [117], which can be used as benchmarking platforms, are very important for the study of device fingerprint analysis and the development of relevant techniques.

As the digital forensic community is gaining more understanding of image device fingerprinting, digital and computational photography has undergone huge development as well. Driven by the need for consumer-level devices to produce better images, we witness significant advances in both hardware and software development. As far as hardware is concerned, the improvement in the design of electronic components like complementary metal-oxide-semiconductor (CMOS) brings better noise immunity. Such improvements allow cameras to have greater flexibility in camera parameter settings, especially for using high signal gain (commonly known by the name of *ISO speed* in photography) without introducing too much noise to images. Thus, digital photography becomes more versatile under different lighting conditions and can be used for high-speed photography. In addition, the ever-increasing computational power of consumer-level mobile devices brought by the improvement in hardware allows more sophisticated computational photography algorithms to be processed in real-time. Among these algorithms, merging multiple time-sequential image frames is a very popular computational photography strategy used by consumer-level devices, especially for *high dynamic range* (HDR) imaging [118]. By processing a burst shots of images, the resultant image can be of higher dynamic range, less noisy and often aesthetically more appealing. Thus, the HDR imaging mode has received great popularity and become available in most mobile imaging devices.

While the above mentioned improvements are beneficial to the users, new challenges are faced by existing PRNU-based device fingerprinting methods. Often, existing PRNU-based device fingerprinting methods are working on the correlation between the noise residuals extracted from the images. The intra-class correlations (the correlations between noise residuals of images from the same source device) can be greatly affected by images' ISO speeds and the alignment operation used in multi-frame computational photography algorithms. This results in compromised forensic accuracy when running existing PRNU-based methods on these images. Thus, insightful investigations are required to understand the problems behind and develop effective forensic methods accordingly. However, the images of the existing datasets in the public domain are not purposefully collected to help answer these problems.

Therefore, we have built a new dataset called *Warwick Image Forensics Dataset*, which can not only serve the same purposes as the existing datasets, but also includes images with their source cameras working in different exposure settings. With the diverse exposure settings, not only systematic investigations in the exposure parameter settings' impact on digital image forensics allowed to be conducted, but also it provides a platform to study different multi-frame computational photography algorithms, which could not be done with the existing forensic datasets. Thus, the dataset paves the way for finding methods to deal with the impact on the accuracy of device fingerprinting due to exposure parameter settings and multi-frame computational photography algorithms.

3.1.1 ISO Speed's Impact On PRNU-Based Digital Forensics

As introduced in Chapter 2, the correlation predictor from [23] is built by considering the following sensor output model as:

$$\mathbf{I} = g^{\gamma} \cdot [(\mathbf{1} + \mathbf{K})\mathbf{Y} + \mathbf{\Lambda}]^{\gamma} + \mathbf{\Theta}_q \tag{3.1}$$

where g is the camera gain, γ is the gamma correction factor and $\mathbf{Y}$ is the scene light intensity. The model considers two major noise terms, represented by $\mathbf{\Lambda}$ and $\mathbf{\Theta}_q$, respectively. $\mathbf{\Lambda}$ is a combination of noise sources including dark current, shot noise and the read-out noise. $\mathbf{\Theta}_q$ represents the quantization noise. The PRNU term of our interest is represented by $\mathbf{K}$, showing the non-uniform response to the scene light intensity $\mathbf{Y}$. The model is simplified in [23] by exploiting the Taylor expansion of the gamma correction and can be written as:

$$\mathbf{I} \doteq \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta} \tag{3.2}$$

with $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$, being the sensor output in the absence of noise, and $\mathbf{\Theta} = \gamma\mathbf{I}^{(0)}\mathbf{\Lambda}/\mathbf{Y} + \mathbf{\Theta}_q$, being a complex of PRNU-irrelevant random noise components. Written in this form, the PRNU component $\mathbf{I}^{(0)}\mathbf{K}$ is a multiplicative term with the noise free image $\mathbf{I}^{(0)}$. While this expression can make people easily miss the role of camera gain, g, in the sensor output model, we still can make some qualitative observations on the impact from camera gain on PRNU. Given similar $\mathbf{I}^{(0)}$ from different images, the size of $\mathbf{\Theta}$ would differ with different camera gain g as higher g requires less input intensity $\mathbf{Y}$ to produce the same output signal $\mathbf{I}^{(0)}$. As $\mathbf{\Theta} = \gamma\mathbf{I}^{(0)}\mathbf{\Lambda}/\mathbf{Y} + \mathbf{\Theta}_q$, a smaller $\mathbf{Y}$ will induce more PRNU-irrelevant noise in an image's noise residual. Because PRNU is often estimated as the noise residual of an image, the addition of PRNU-irrelevant noises will make this image's noise residual less correlated with noise residuals extracted from other intra-class images (images from the same source device).

With the above relationship in mind, in [119], the authors empirically show that given similar contents in images taken with different ISO speed settings, the intra-class correlation distributions can vary according to ISO speeds, which directly control the camera gain g. This results in higher error rates in source camera identification for images of higher ISO speeds. Due to this phenomenon, [119], an empirical study on how ISO speed would affect PRNU-based source camera identification, suggests that camera exposure parameters like ISO speed should be considered from a forensic perspective. It is also suggested that the construction of forensic image datasets should include images of different exposure parameter settings, which can also be beneficial for studies in steganalysis.

3.1.2 High Dynamic Range Imaging

HDR images can capture more details from scenes compared to standard dynamic range (SDR) images and hence receive much attention from computational photography researchers. From the early works in [120, 121] to the more recent works like HDR+ [122] and deep neural network based methods [123], different HDR imaging techniques are developed to allow them to be used under different conditions. Despite the differences, these methods also share a few things in common, which make HDR images a hard subject in general for PRNU-based device fingerprinting. For most HDR imaging algorithms, conventional exposure methods of taking a set of time-sequential images are often used, despite some methods have images with the same exposure time and some others use images with different exposure time. A radiance map can be reconstructed from a set of time-sequential images and provides a larger dynamic range than single exposure images. However, as it is almost impossible to avoid object or camera motion during the capturing process of the time-sequential image sets, the reconstruction of the radiance map usually involves pixel-wise alignment to compensate the object motions across different image frames to avoid motion blurring. Such an operation will mix the PRNU signal from different pixel and cause misalignment between the PRNU embedded in the resultant HDR images and reference PRNU extracted from single exposure images taken by the same camera. Due to such misalignment, intra-class PRNU pairs will be less correlated and cause difficulty in PRNU-based provenance analysis.

In addition to the misalignment problem, tone mapping is another operation commonly used in HDR algorithms, which can cause trouble for existing PRNU-based forensic methods. Tone mapping is used to reconstruct a color image from a radiance map. Each implementation of different HDR algorithms may have its unique tone mapping curve and on top of that, different tone

mapping curves can be applied either globally or locally on the same image. As mentioned in Chapter 2, PRNU-based forgery localization methods often use a content dependent correlation predictor to estimate the block-wise intra-class correlations to discover pixels with its PRNU absent, without the prior knowledge of the tone mapping curve, reliable predictions from the correlation predictor can hardly be expected. These problems require specific adjustment for existing PRNU-based methods to make them effective on HDR images.

3.1.3 Existing Public Image Datasets

As a rapidly developing topic, device fingerprinting draws many researchers' attention and several image datasets are constructed over the years to facilitate the researches. One of the earliest image datasets adopted for device fingerprinting is the Uncompressed Colour Image Dataset (UCID)[124]. From then on, more dedicated image datasets for provenance analysis are constructed. Notably, the Dresden Image Dataset [116], RAISE dataset [125] and VISION dataset [117] are three datasets widely used for benchmarking in device fingerprinting. Each dataset consists of a large number of high resolution images from multiple devices, either digital cameras or smartphone cameras. More recent datasets like the SOCRatES [126] and DAXING datasets [127] feature images from a vast number of source devices (103 smartphone cameras from SOCRatES and 90 smartphone cameras from DAXING dataset). Despite the images from these datasets show good diversity and heterogeneity in terms of contents, all the above mentioned datasets focus on SDR images only and the diversity in camera exposure parameter settings was not given adequate consideration during the construction of these datasets.

The 'HDR dataset' from [128] is the first forensic dataset featuring HDR images. The images in this dataset are taken with 23 smartphone cameras and for each scene included in this dataset, both a SDR image and a HDR image are provided. The images are taken under three different conditions: taken from the tripod, by the hand and by a shaky hand. Despite [128] featuring both SDR and HDR images, its real contribution of the image pairs towards the understanding of HDR images' impact on source device identification is limited. Firstly, the SDR images included in the dataset are not the SDR images used for the construction of the HDR images. As a result, these pairs may not best reflect the impact of HDR algorithms on device fingerprints in SDR images. Secondly, as the HDR images in this dataset are generated directly from the smartphones, the coverage of different implementations of HDR algorithms are confined by the choice of smartphones included in this dataset. As the development of new HDR algorithms continues, research findings stemmed from this dataset are unlikely to be applicable to other

HDR images produced by future algorithms. Acknowledging this problem, our Warwick Image Forensics Dataset takes the flexibility of generating HDR images using different implementations of HDR algorithms into account as we shall see from the following section.

The details of the datasets mentioned above are summarized in Table 3.1.

3.2 Dataset Details

In this section, we present the details of our Warwick Image Forensics Dataset.

3.2.1 The Selection of Cameras

The images from the Warwick Image Forensics Dataset are captured by 14 digital cameras. The details and the technical specifications of the cameras are shown in Table 3.2. The primary goal of this dataset is helping the digital forensic community to develop better understanding of the impacts from both camera exposure parameter settings and multi-frame computational photography algorithms, especially HDR imaging, on device fingerprinting. The choice of using digital cameras instead of smartphone cameras in this dataset allows us to have better control on camera exposure parameter settings during the image capturing process. With these fine controls, the images captured are suitable for different HDR algorithms, whether they are using images of the same or different exposures to produce HDR images. The 14 cameras are from 11 different models and cover a good range of major camera manufacturers. Also, the 14 cameras show good diversity of different image sensor formats with the smallest sensor of comparable size to the sensors used on smartphones cameras.

3.2.2 Image Acquisition

The images from this dataset can be categorised into the following three classes:

- Flatfield images
- SDR images
- HDR-ready SDR images

The *flatfield images* are mainly for reference PRNU extraction. For each camera, 100 flatfield images are captured by taking photos of a flat blue board with the lenses adjusted to be out of focus. For each image shot, the camera is set to its lowest ISO speed to reduce the amount of read-out noise in the

Table 3.1: Details of different forensic datasets mentioned in Section 3.1.3, compared with the Warwick Image Forensics Dataset shown at the bottom

Dataset	No. of camera models	No. of devices	No. of images	Equal No. of images at each ISO speed?	HDR images
UCID[124]	1	5	1338	No	N.A.
Dresden[116]	25	73	18,452	No	N.A.
RAISE[125]	3	3	8,156	No	N.A.
VISION[117]	25	35	11,732	No	N.A.
SOCRatES[126]	60	104	9,700	No	N.A.
DAXING[127]	22	90	43,400	No	N.A.
HDR[128]	21	23	5,415	No	HDR images generated directly from camera
Warwick Image Forensics Dataset[3]	12	14	58,600	Yes	SDR images can be adopted by different HDR algorithms to generate HDR images

Table 3.2: Details of the cameras presented in Warwick Image Forensics Dataset

No.	Camera	Resolution	Sensor Format	Sensor Dimensions	CFA Type	Lens
1	Canon EOS 6D	3648×5472	35 mm	35.8×23.9 mm^2	Bayer Filter	Interchangeable
2	Canon EOS 6D Mark II	4160×6240	35 mm	35.9×24 mm^2	Bayer Filter	Interchangeable
3	Canon EOS 80D	4000×6000	APS-C	22.5×15 mm^2	Bayer Filter	Interchangeable
4	Canon EOS M6	4000×6000	APS-C	22.3×14.9 mm^2	Bayer Filter	Interchangeable
5	Fujifilm X-A10_1	3264×4896	APS-C	23.6×15.6 mm^2	Bayer Filter	Interchangeable
6	Fujifilm X-A10_2	3264×4896	APS-C	23.6×15.6 mm^2	Bayer Filter	Interchangeable
7	Nikon D7200	4000×6000	APS-C	23.5×15.6 mm^2	Bayer Filter	Interchangeable
8	Panasonic Lumix DC-TZ90_1	3888×5184	1/2.3”	6.16×4.62 mm^2	Bayer Filter	Fixed
9	Panasonic Lumix DC-TZ90_2	3888×5184	1/2.3”	6.16×4.62 mm^2	Bayer Filter	Fixed
10	Olympus E-M10 Mark II	3456×4608	Four Thirds	17.3×13 mm^2	Bayer Filter	Interchangeable
11	Sigma Sd Quattro	3616×5424	Foveon X3	23.4×15.5 mm^2	NA	Interchangeable
12	Sony Alpha 68	4000×6000	APS-C	23.5×15.6 mm^2	Bayer Filter	Interchangeable
13	Sony RX100_1	3648×5472	1”	13.2×8.8 mm^2	Bayer Filter	Fixed
14	Sony RX100_2	3648×5472	1”	13.2×8.8 mm^2	Bayer Filter	Fixed

image. The exposure metering of each shot is adjusted to normal exposure, making the images neither too dark nor too saturated.

The *SDR images* in this dataset are the standard dynamic range images taken with the cameras' single-shot mode and thus cannot be used for HDR merging algorithms. These images are taken with systematic control of the cameras' ISO speed. For each camera, images are taken with the ISO speed set to be one of the following values: ISO 100, 200, 400, 800, 1600, 3200 and 6400, with the only exceptions from the two Panasonic Lumix DC-TZ90 as their ISO speeds go only up to 3200. 30 images of different scenes in different conditions are taken for each above mentioned ISO speed on each camera. For each image shot, with the camera's ISO speed set, we enable the camera's *Program Mode*, allowing the camera to adjust its aperture size and exposure time automatically to allow sufficient exposure. Almost all the images from this set are taken in a hand-held style. This set of images provide good diversity in scenes as well as camera exposure parameter settings at the same time.

The *HDR-ready SDR images* are the set of standard dynamic range images, which can be used with different algorithms to produce HDR images. Images of 20 different scenes are taken for this set. Different HDR algorithms may require different sets of images. For example, [121] uses set of images of varying exposure times and [122] expects a burst shot of under-exposure images with the same exposure time, we took continuous shots of images using three different modes. The first one is using the auto exposure bracketing (AEB) function on each camera. The AEB function allows us to take continuous shots of images with varying exposure times. The second and third modes both use fast continuous shot mode to take at least 7 continuous shots of images with the same exposure. However, one set is taken at normal exposure and the other is taken as under-exposed, usually by 1 or 2 stops measured by the cameras' exposure metering system. An example of the images taken with these three modes are shown in Fig. 3.1. Furthermore, to increase the diversity in exposure parameter settings, we systematically repeat these three modes with cameras set to 7 different ISO speeds as mentioned above. Thus, for each camera, more than 120 images of the same scene with various camera parameter settings are taken. The 20 different scenes included in this dataset are carefully selected, covering both indoor and outdoor, day-light and night environment, still and dynamic scenes as well as objects with different texture. The images are taken with the cameras either hand-held or sat on a tripod. The dataset does not provide any generated HDR images from a specific HDR algorithm directly as it would not be useful considering the existence of different HDR algorithms. Instead, the good diversity of camera exposure parameter settings in this dataset provides the users with the flexibility of adopting different HDR imaging algorithms for the images, allowing the research findings to

be more generic but not just limited to a specific HDR algorithm. Also, the good diversity of camera exposure parameter settings means that the images from this dataset can be used for other camera exposure parameter setting dependent studies.

For every image from our Warwick Image Forensics Dataset, both the unaltered RAW image file and the camera generated JPEG image file are available.

3.3 Experimental Evaluations

In this section, we conduct experimental evaluations on PRNU-based source camera identification and clustering's performance on the Warwick Image Forensics Dataset. In particular, we will show how the performance varies by using images of different ISO speeds for the tests.

For source camera identification, from each camera, we extract the reference PRNUs from 100 flatfield JPEG images using the BM3D de-noising algorithm [129]. The extracted reference PRNUs are processed by a spectrum equaliser from [130] to remove unwanted artefacts. We test the performance of source camera identification method from [20] on the SDR images from the dataset. For each image, we crop a region of 512×512 pixels from its center to extract the noise residual and compute the correlations with the corresponding pixels from the reference PRNUs. The receiver operator characteristics (ROC) curves for the method on images of ISO speed 100, 200, 400, 800, 1600 and 3200 are shown in Fig. 3.2. Apparently, as the ISO speed gets higher, smaller under curve area is observed indicating worse performance. Fig. 3.3 shows the correlation matrices of pairwise correlations between noise residuals extracted from SDR images of ISO speed 100, 200, 400, 800, 1600 and 3200. On the plots, we use red squares to highlight the intra-class correlations belonging to each camera, marked by the number which follows the order in Table 3.2. The six color-maps follow the same color scheme as shown in the bar on the right. The cluster structures in each plot become less clear as the ISO speed gets larger. The clustering performance show the same general trend with smaller F1 score for the higher ISO speed despite the F1 score for ISO 200 images is slightly higher than the one for ISO 100 images. By applying the method from [25], we have F1 score of 84.33%, 84.51%, 83.12%, 82.86%, 80.97% and 80.13% for ISO speed 100, 200, 400, 800, 1600 and 3200, respectively.

All experiments mentioned above prove that different camera exposure settings have different levels of impact on the quality of PRNU and the forensic analyses, which need to be considered in forensic research and real-world investigations. Therefore, it is important to include images of diverse camera parameter settings in the image datasets in order to facilitate researches.

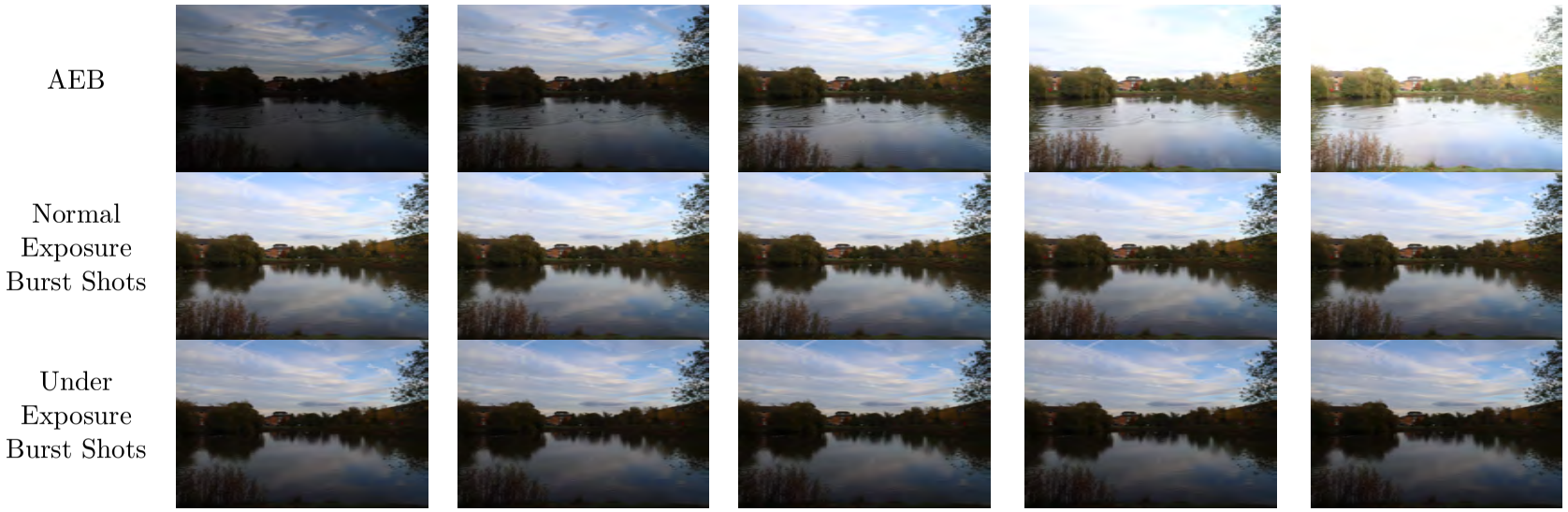


Figure 3.1: Sample images of a scene from the *HDR-ready SDR images* in Warwick Image Forensics Dataset. These images are taken by a Canon EOS 6D Mark II with ISO speed set to 100. From top to bottom, we show the images taken with three different modes. The top one uses the camera's auto exposure bracketing (AEB) function and the following two rows are shots with consistent exposure time within each row. The middle row has normal exposure and the images in the bottom row are under exposed by 1 stop measured by the cameras exposure metering system. Due to the limit of space, we only show a portion of the images taken with three modes at ISO 100.

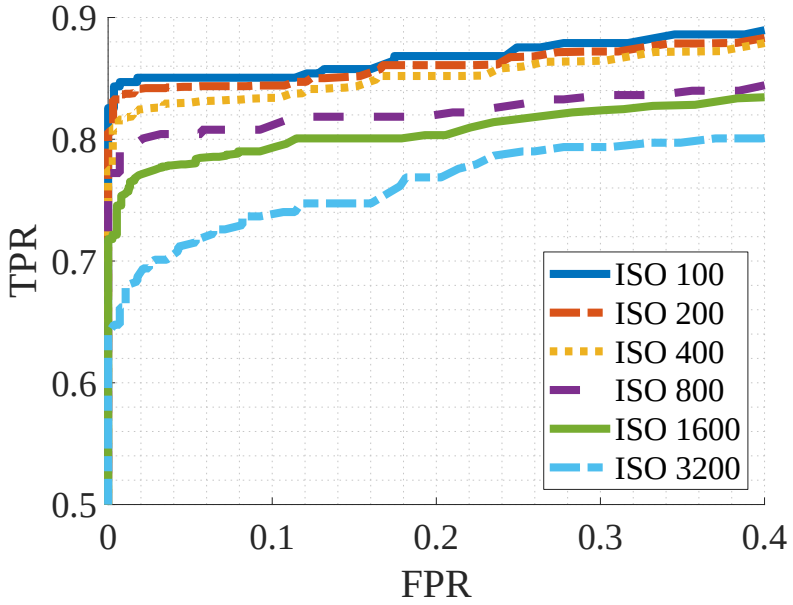


Figure 3.2: The ROC curves of source camera identification using the method from [20] on SDR images with ISO speed 100, 200, 400, 800, 1600 and 3200.

3.4 Conclusion

In this chapter, we demonstrated the impact of camera exposure parameter settings like ISO speed on the quality of PRNU and the importance of having an image dataset that can facilitate future research into the development of better solutions to deal with this impact. We presented the Warwick Image Forensics Dataset, a novel forensic image dataset consisting of more than 58,600 images, captured with special attentions to exposure parameter settings. The images are from 14 different digital cameras. The good diversity of camera parameter settings allows studies on different exposure parameters' impact on device fingerprinting to be carried out on this dataset. With the diverse ways of taking these images, they can easily be used by different multi-frame computational photography algorithms including HDR imaging. Thus, HDR image related studies in device fingerprinting can be carried out using this dataset as well. In addition, the dataset can also be used for other studies like steganalysis. Thus, we believe it is beneficial for the digital forensic community with the dataset released as an open-source.

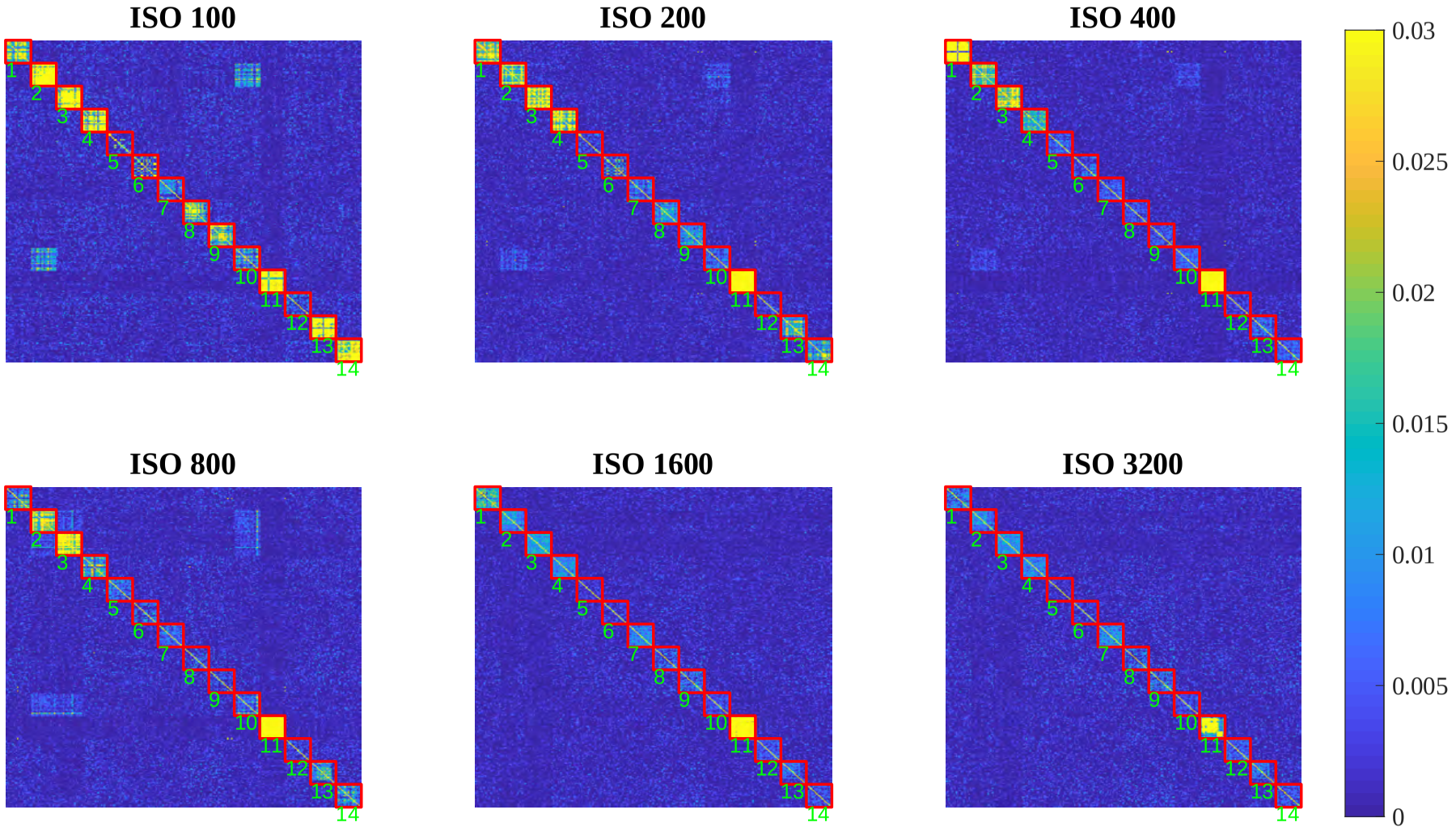


Figure 3.3: Correlation matrices for the pairwise correlations between SDR images of ISO speed 100, , 200, 400, 800, 1600 and 3200.

Chapter 4

Impact of ISO Speed upon PRNU and Forgery Detection

With the Warwick Image Forensics Dataset constructed, this platform allows more detailed investigations into the ISO speed's impact on PRNU-based image forgery detection to be carried out. Hence, we use this chapter to present the work done on this topic. We will use Section 4.1 to further explain the background. The analysis of the impact of ISO speed on PRNU-based image forgery detection and the proposed methods to mitigate this impact are presented in Section 4.2, 4.3 and 4.4. Specifically, we first analytically and empirically prove in Section 4.2 that the correlation between an image's noise residual and its reference PRNU is not only content-dependent as previously known, but also dependent on the *camera sensitivity setting* (i.e. the *ISO speed*). We then validate our postulate in Section 4.3 that, due to such ISO speed dependency, reliable predictions of the correlation between an image's noise residual and its reference PRNU can only be accurately made when a correlation predictor is trained on images of similar ISO speeds to the image in question. Base on the postulate, we propose an ISO specific correlation prediction process. Recognizing that in the real-world, information about the ISO speed may not be available to facilitate the implementation of our postulate in the correlation prediction process, we propose a method called Content-based Inference of ISO Speeds (CINFISOS, /'sin.fə.səs/) in Section 4.4 to infer the ISO speed from the image content. Comprehensive experiments to test the proposed CINFISOS and the ISO specific correlation prediction process for forgery detection are presented in Section 4.5. Section 4.6 concludes this chapter.

4.1 Introduction

Photo Response Non-Uniformity (PRNU) based methods have shown their unique strength in image forgery detection. Many different algorithms have been proposed for PRNU-based image forgery detection [21, 23, 24, 83, 84, 86]. In most of these works, PRNU is utilised by computing the image-wise or block-wise correlations between the source device's reference PRNU and the test image's PRNU. The corresponding pixel-wise decision (forgery detection) can be made by comparing the correlations with a decision threshold.

As mentioned in the previous sections, the PRNU is often estimated in the form of the noise residual of an image, which can be extracted from an image by simply subtracting the de-noised image from the original image. By nature, PRNU is a weak noise. The existence of camera artifacts and other PRNU-irrelevant noises (e.g. shot noise, thermal noise, etc.) in an image's noise residual can reduce the correlation between the noise residual and the device's reference PRNU. It becomes a non-trivial problem to separate the inter-class (images from different source devices) from the intra-class (images from the same source device) correlations. It becomes particularly problematic when the PRNU quality in the noise residual is poor such that these two types of correlations' distributions can have large overlaps.

Despite a large number of works that have been done to better extract, estimate and enhance the PRNU [22, 23, 131–135], the overlap between inter- and intra-class correlations cannot be completely avoided. Thus, many researchers have been working on refining the choice of the decision thresholds to better separate the two classes, especially for image forgery detection [23, 24, 86]. The decision thresholds are often set with reference to the expected intra-class correlations predicted by a correlation predictor. The correlation between an image's noise residual and the device's reference PRNU reflects the strength of the PRNU in the image. As the strength of the PRNU is multiplicative of the pixel intensity and some highly textured image content or post-processing may damage the PRNU's quality, correlation prediction should be performed in an adaptive manner. A content-dependent correlation predictor is proposed by Chen *et al.* in [23], which formulates the correlation predictor as a regressor model of four image features, namely the *intensity*, *texture*, *signal-flattening* and a *texture-intensity combinative term*. This correlation predictor has been adopted by many PRNU-based forgery detection algorithms (e.g. [23, 24, 83, 84, 86]). Due to the complex nature of the PRNU correlation, despite different attempts to re-engineer the correlation predictor over the past decade, we have not witnessed much success. Thus, the digital forensic community still relies greatly on the correlation predictor from [23] for PRNU-based forgery detection.

However, over the last decade, we have also witnessed great advancement in the digital camera industry, especially in sensor design. Such advancement also brings new challenges to PRNU-based digital forensics. Therefore, we have observed a few issues about the correlation predictor proposed in [23]. An important feature ignored by the correlation predictor is the camera sensitivity setting, which is commonly known by the name of ISO speed. Many camera manufacturers have been working on improving sensor performance and providing more and higher ISO speeds to digital cameras. It allows the photographers to take photos under different lighting conditions. While the improvements have been brought to sensor technology, it is also a known fact that high ISO speeds may introduce more noise to an image. As a result, the quality of the PRNU left in the noise residual will be reduced when a high ISO speed is used. [119] empirically shows that source camera identification performance could be degraded for images taken at higher ISO speeds, which is further validated in Chapter 3 by testing source camera identification performance at multiple ISO speeds. With camera manufacturers increasingly supporting broader ranges of ISO speed settings on digital cameras and mobile devices, a proper analysis of the ISO speed's influence on PRNU-based image forensics, especially on the correlations and image forgery detection, needs to be carried out.

As this chapter focuses on the correlation between an image's noise residual with its reference PRNU, for simplicity, we will call it the *correlation*.

4.2 ISO Speed Dependent Correlation

In this section, we demonstrate that an image's ISO speed can affect its correlation. As a general noise model can be complicated, to show the existence of such an *ISO Speed-Correlation* relationship in a concise manner, we use a special case to prove this relationship analytically and then empirically show it with more general cases. The special case considered is a single color channel of a flat-field RAW image, from which we expect the same value for every pixel if they are noise-free. To conduct PRNU-based pixel-wise forgery detection, the correlation between the noise-residual of a block centered at each pixel and the corresponding block of the reference PRNU is calculated. Let $\boldsymbol{z}$ be a noise residual within a block N_i centered at pixel i and $\boldsymbol{\omega}$ be the reference fingerprint within the corresponding block. Assume both $\boldsymbol{z}$ and $\boldsymbol{\omega}$ are standardised, which means they follow the normal distribution $\mathcal{N}(0,1)$. We can model both signals as the sum of a PRNU component and a PRNU-irrelevant part. At pixel $j \in N_i$:

$$\begin{cases} \omega_j = x_j + \alpha_j \\ z_j = y_j + \beta_j \end{cases} \tag{4.1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the PRNU components of $\boldsymbol{\omega}$ and $\boldsymbol{z}$ while $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the PRNU-irrelevant noises. As for a flat-field image, we can approximate its PRNU component, $\boldsymbol{x}$ in this case, as a normal distribution $\mathcal{N}(0, {\sigma_x}^2)$ and $\boldsymbol{\alpha}$ conforms to $\mathcal{N}(0, 1 - {\sigma_x}^2)$. For intra-class pairs, $\boldsymbol{x}$ and $\boldsymbol{y}$ represent the same PRNU. As they may differ in strength, without losing generality, we can express $\boldsymbol{y}$ as $\mathcal{N}(0, {\sigma_y}^2)$ with $\sigma_y = \sqrt{\lambda}\sigma_x$ and $\boldsymbol{y} = \sqrt{\lambda}\boldsymbol{x}$. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are mutually independent. So when we compute the correlation ρ_i of the block N_i, the correlation ρ_i becomes:

$$\rho_i \sim \mathcal{N}(\mu_i, \Sigma_i) \tag{4.2}$$

with

$$\begin{cases} \mu_i = \sigma_x \sigma_y = \sqrt{\lambda}{\sigma_x}^2 \\ \Sigma_i = (1 + \lambda {\sigma_x}^4)/|N_i| \end{cases} \tag{4.3}$$

From the above expression, we can see that the expected correlation value, μ_i, is proportional to the standard deviation σ_y of the PRNU component, $\boldsymbol{y}$, in the image's noise residual, $\boldsymbol{z}$. Based on the Poissonian-Gaussian noise model [88, 136, 137], we can see that the ISO speed would affect this standard deviation σ_y and eventually exert influence on the PRNU correlations.

The relationship between the camera gain, g, which is directly determined by the camera's ISO speed, and the noisy raw pixel intensity, I, is analysed in [88]. The raw pixel intensity is proportional to the number of electrons counted on the sensor. Photo-electron conversion is the main source of the electrons collected from the sensor. [88] considers the Poissonian statistics of the incident photon counting process as follows. At pixel i, the number of the counted electrons is the sum of the electrons generated from photo-electron conversion N_{p_i} and dark electrons N_{ti} from the thermal noise. It is assumed that the variance of the thermal noise is uniform across the sensor and all other electronic noises can be modelled as a zero-mean Gaussian noise with variance s^2. So the raw pixel intensity, I_i, at pixel i, can be written as:

$$\mathrm{I}_i \sim g \cdot [p_0 + \mathcal{P}(\eta_i N_{p_i} + N_{ti} - p_0) + \mathcal{N}(0, s^2)] \tag{4.4}$$

where $\mathcal{P}(\cdot)$ represents the Poisson distribution and η_i is the photon-electron conversion rate at pixel i. p_0 is a base pedestal parameter introduced in the camera design to provide an offset-from-zero of the pixel's output intensity. For each pixel, as a large number of electrons are counted, the normal approximation of Poisson distribution can be exploited. Therefore, I_i can be modeled as:

$$\mathrm{I}_i \sim \mathcal{N}(\varphi_i, g\varphi_i + t) \tag{4.5}$$

with

$$\begin{cases} t = g^2 s^2 - g^2 p_0 \\ \varphi_i = g \cdot (\eta_i N_{p_i} + N_{ti}) \end{cases}, \tag{4.6}$$

φ can be viewed as the expected pixel intensity. Notice that this model from [88] has not yet considered the PRNU. To include the PRNU in this model, we write the photo-electron conversion rate η_i as the following expression by considering the non-uniform response of each pixel to the photons:

$$\eta_i = \bar{\eta}(1 + k_i), \tag{4.7}$$

where $\bar{\eta}$ is the average photo-electron conversion rate and k_i is the PRNU factor at pixel i. k follows normal distribution $\mathcal{N}(0, \sigma_k^2)$. As we are considering the case of a flat-field image here so we can fix the number of photons, N_{p_i}, collected at every pixel. By expanding Equation (4.5), we have:

$$\mathrm{I}_i \sim \mathcal{N}((1 + k_i)\varphi - g k_i N_{ti}, g(1 + k_i)\varphi + t - g^2 k_i N_{ti}) \tag{4.8}$$

As in most cases, both the PRNU and the thermal noise are weak noises. We can ignore the terms involving $k_i N_{ti}$. When we consider a block N_i, often it consists of thousands of pixels (e.g. 4096 pixels for a 64×64 block). Such a large number of pixels allow us to approximate the overall distribution of the pixel values in this block by another normal distribution. By substituting t of Equation (4.8) with the expression for t in Equation (4.6), we approximate the distribution of the pixel values in block N_i as:

$$\mathrm{I}_{N_i} \sim \mathcal{N}(\varphi, \varphi^2 {\sigma_k}^2 + g\varphi + g^2 s^2 - g^2 p_0) \tag{4.9}$$

We expect the de-noised version of this block to have pixels of uniform intensity, φ. Thus, we can approximate the variance of the noise residual of this block as:

$${\sigma_{\mathrm{res}}}^2 \approx \varphi^2 \sigma_k^2 + g\varphi + g^2 s^2 - g^2 p_0 \tag{4.10}$$

The PRNU component in the noise residual has a variance of $\varphi^2 \sigma_k^2$. By normalizing the noise residual, the standard deviation of the PRNU component in the normalized noise residual becomes:

$$\sigma_y = \sqrt{\frac{\varphi^2 \sigma_k^2}{\varphi^2 {\sigma_k}^2 + g\varphi + g^2 s^2 - g^2 p_0}} \tag{4.11}$$

Clearly, σ_y is dependent on the camera gain g. By substituting this expression back to Equation (4.3), we can conclude that the correlation ρ_i can be affected by the camera gain g and thus affected by ISO speed.

Notice that when we introduce PRNU by considering different photo-

electron conversion rate, η_i, at each pixel to the raw pixel intensity model from [88], the noise residual variance model described in Equation (4.10) becomes a quadratic function of the expected pixel intensity φ, which can be expressed as:

$$\sigma_{\text{res}}{}^2 = A\varphi^2 + B\varphi + C \tag{4.12}$$

with

$$\begin{cases} A = \sigma_k{}^2 \\ B = g \\ C = g^2 s^2 - g^2 p_0 \end{cases} \tag{4.13}$$

It differs from the linear model in [88]. We will empirically validate Equation (4.10) to show the physical importance of the PRNU term, $\varphi^2 \sigma_k{}^2$, in the equation despite the approximations made.

We use four cameras for the test, namely a Nikon D7200, a Canon 6D MKII, a Canon 80D, and a Canon M6. Each of the four cameras can generate 14-bits RAW images, which means their pixel values can vary between the range of $[0, 16383]$. To better show the physical meaning of the coefficients in Equation (4.10), we standardise the pixel values to the range of $[0, 1]$. To validate Equation (4.10), we plot the variance of the noise in the flat-field images against different pixel values in Fig.4.1. We use the cameras to take images of a screen of flat color to make the captured images as plain as possible to avoid the interference from image content. Each camera's ISO speed is set to 100. The exposure time is varied to change the pixel intensity for different shots. As the cameras use Bayer-filter as their color filtering array (CFA), we subsample the RAW images with a stride of 2 in both vertical and horizontal directions to make sure the pixels we test are from the same color channel. Despite the set-up, the images are not completely flat due to other camera artifacts, e.g. vignetting. Thus, we use the method from [88] to estimate the expected pixel value and variance for multiple image blocks from each noisy RAW image. Fig.4.1 shows the fitting of Equation (4.12) to the experiment data, which is computed using ordinary least squares (OLS). Despite the noisy nature of the data, the large correlation coefficients (r^2) for the plots show that a good agreement between the model and the data can be observed.

In addition to showing the good agreement of the derived model and the real data, we would like to show the physical meaning of the first order coefficient, $B = g$ in the model as well. We use the RAW images from the same Canon 6D MKII from the previous test for this test. We repeat the previous experiment four times but set the cameras' ISO speed to ISO 200, 400, 800, and 1600, respectively. Again, we fit Equation (4.12) to the data. As for the same camera, despite the change of ISO speed, we can assume that the PRNU factor on the sensor should remain the same and so does the variance of the PRNU

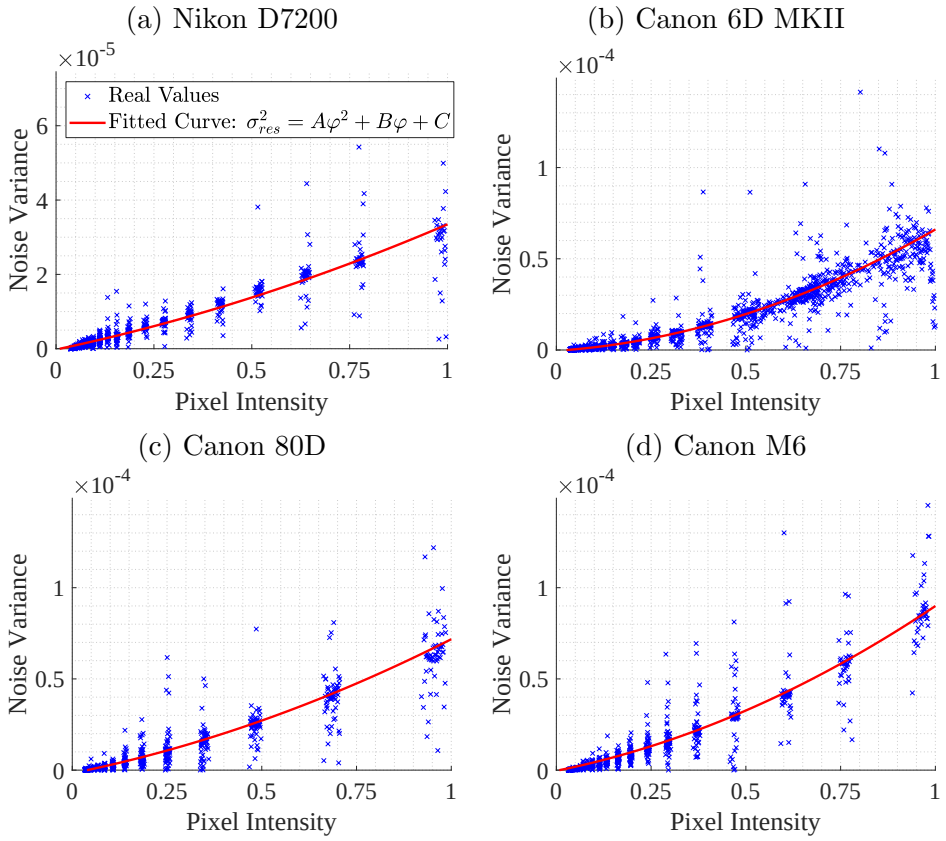


Figure 4.1: Plots of noise's variance ${\sigma_{\text{res}}}^2$ against pixel intensity φ, with a quadratic fitting (red curve) as described by Equation (4.10) and (4.12), of RAW flat-field ISO 100 images from four cameras: (a) Nikon D7200, (b) Canon 6D MKII, (c) Canon 80D and (d) Canon M6. The fitted coefficients for Equation (4.12) and the correlation coefficient (r^2) for each plot are shown in Table 4.1.

factor, σ_k^2. Thus, it is reasonable for us to fix the second order coefficient $A = \sigma_k^2$ to 5.24×10^{-5}, the value estimated from Fig.4.1, in Equation (4.12) for these fittings and the corresponding fittings generated using OLS are shown in Fig.4.2. Once again, good agreement between the fitted curve and the data can be observed with the large correlation coefficients. In addition, we show a $\log - \log$ plot of the estimated first order coefficients B from Fig.4.1(b) and 4.2 against the ISO speed of their corresponding images in Fig.4.3. We fitted a straight line to the plot given slope close to 1. As a camera's ISO speed is proportional to its camera gain, g, the straight line with slope close to 1 in Fig.4.3 shows that B and g follow a linear relationship, which validates our noise model from Equation (4.10) with $B = g$. Therefore, it confirms that the correlation model is dependent on ISO speed.

The above conclusions are made for the special condition when we consider the images to be RAW flat-field image. When we take post-processings (e.g. color interpolation and JPEG compression) and the influence due to the image

Table 4.1: The fitted coefficients for Equation (4.12) and the correlation coefficient (r^2) for each plot shown in Fig. 4.1.

	A	B	C	r^2
Nikon D7200	1.14×10^{-5}	2.23×10^{-5}	-2.20×10^{-7}	0.8747
Canon 6D MKII	5.24×10^{-5}	1.41×10^{-5}	-4.33×10^{-7}	0.8967
Canon 80D	3.15×10^{-5}	4.20×10^{-5}	-1.70×10^{-6}	0.8377
Canon M6	4.85×10^{-5}	4.18×10^{-5}	-3.51×10^{-7}	0.8263

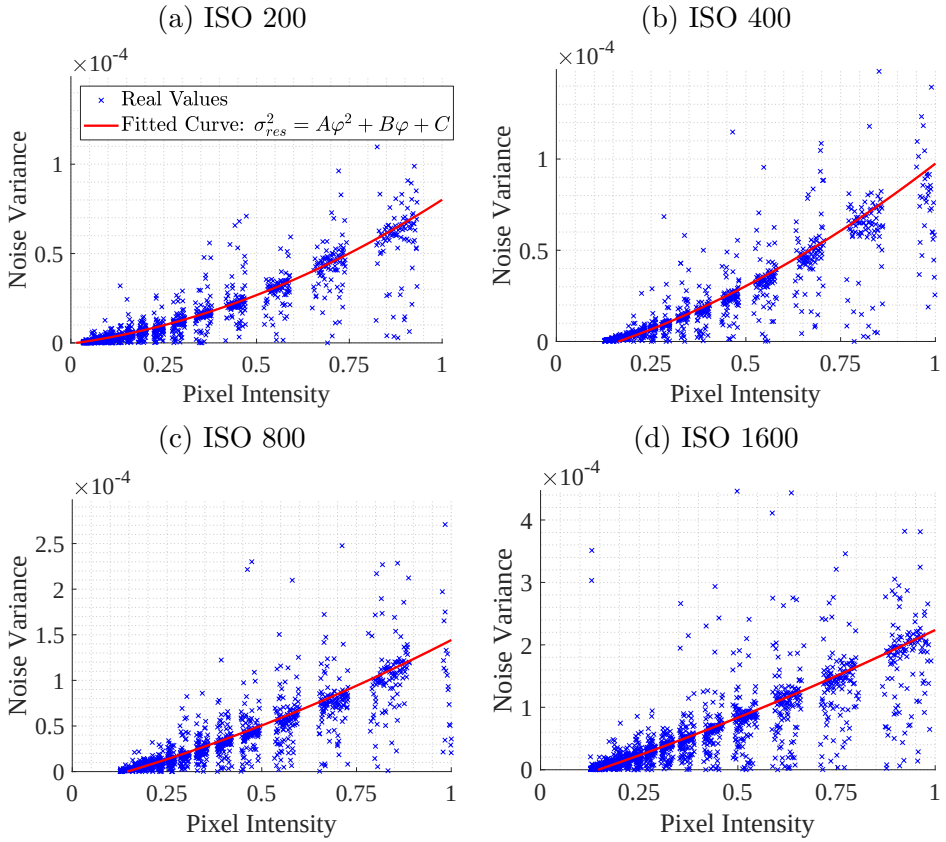


Figure 4.2: Plots of noise's variance ${\sigma_{\text{res}}}^2$ against pixel intensity φ of images with different ISO speed from a Canon 6D MKII. We fit Equation (4.10) to the plots with a fixed second order coefficient, $A = \sigma_k^2 = 5.24 \times 10^{-5}$, estimated from Fig.4.1(b). The first order coefficient B and the correlation coefficients for the four fittings are shown in Table 4.2.

Table 4.2: The fitted first order coefficient B and the correlation coefficients for the four fittings shown in Fig. 4.2.

	B	r^2
ISO 200	2.81×10^{-5}	0.8605
ISO 400	5.56×10^{-5}	0.8172
ISO 800	1.09×10^{-4}	0.7503
ISO 1600	2.02×10^{-4}	0.7516

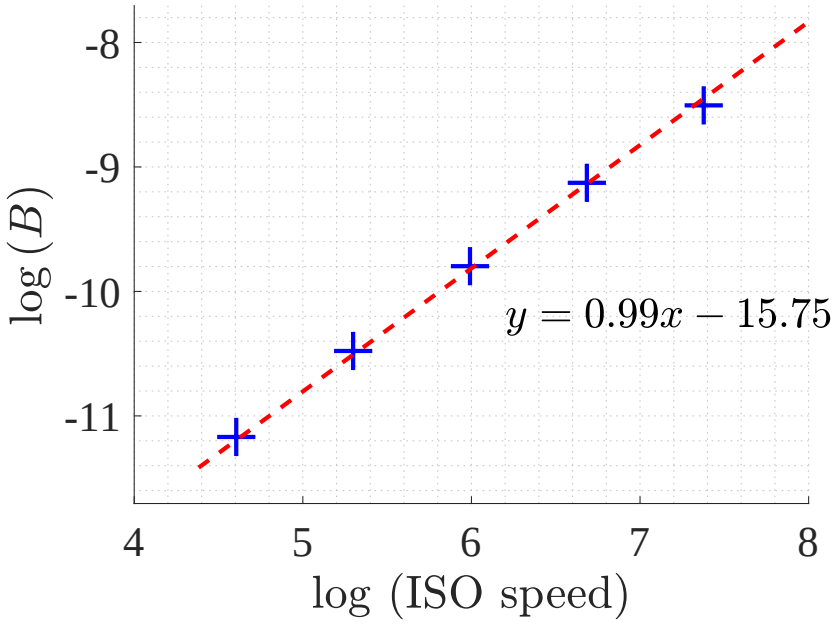


Figure 4.3: log-log plot of the estimated first order coefficient B against the ISO speeds of the images used to estimate B. A straight line is fitted with a slope of 0.99

content into consideration, the noise model could become rather complicated. This is both because the PRNU is multiplicative of image content and image content may propagate into the noise residual due to imperfect denoising. Actually, higher ISO images are more likely to suffer from strong JPEG compression and imperfect denoising (see Appendix A). Thus, though Equation (4.10) cannot be translated directly to the general conditions, all the factors suggest a higher ISO speed can introduce more PRNU-irrelevant noise. As a result, this will reduce the proportion of signals corresponding to the PRNU in the noise residual and eventually reduce the correlation. We use Fig.4.4 to empirically show that the correlation is dependent on the image's ISO speed when post-processings such as de-mosaicing, gamma correction, JPEG compression, etc., are applied to a non-flat RAW image.

The images shown in Fig.4.4 are from a Canon 6D MKII camera in the Warwick Image Forensics Dataset. All the images shown here are saved in the JPEG format by the camera's default setting. Images of two scenes are taken under different ISO speeds using different exposure times to ensure that every image can reach the same exposure level. Thus, there is nearly no difference in pixel intensity between the images of the same scene. As the PRNU is a multiplicative signal, having images of the same pixel intensity of the same image content allows us to make a fair comparison with ISO speed's impact on the correlation. The correlation heat maps in Fig.4.4 are computed by correlating the noise residuals from the images' green channel with the device's

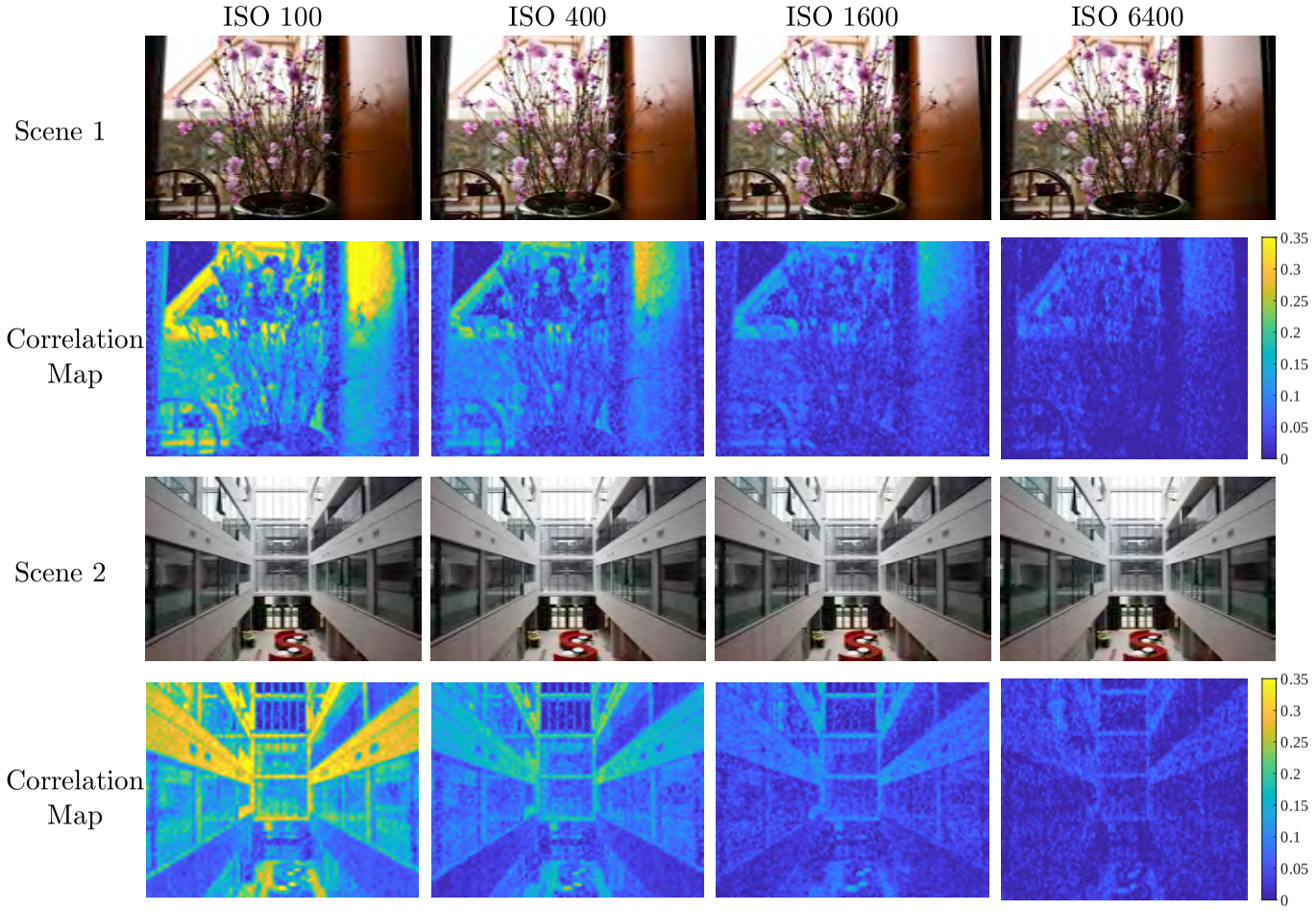


Figure 4.4: Image of two different scene from a Canon 6D MKII from the Warwick Image Forensics Dataset. The images are taken with different ISO speeds. The exposure time for each image is set accordingly to let the images of the same scene have similar exposure level. The block-wise correlation maps are computed with a block size of 128×128 pixels. The color bars used for the correlation maps are at the right hand side, next to the ISO 6400 correlation maps.

reference green channel PRNU. The reference PRNU is extracted from 50 flat-field images. The block size for the computation of the correlation at each pixel is 128×128 pixels. We use yellow to show high correlation regions and blue to show the opposite. Apparently, as the ISO speed increases, the correlation map shows more regions with low correlation. It can be concluded that despite these images with complex image content have undergone post-processing, their correlation with the reference PRNU is still dependent on the image's ISO speed.

4.3 ISO Speed's Impact Upon Correlation Prediction

A correlation predictor is an important component of many PRNU-based tampering localization methods. Many PRNU-based tampering localization methods are applied by comparing the block-wise correlations with a decision threshold set according to the predicted correlation. As a result, the choice of the decision threshold and the performance of these methods can be greatly affected by the accuracy of the correlation prediction. As the correlation is content dependent, without considering the ISO speed, [23] models the correlation as a function of four image features, namely the intensity, texture, signal flattening and a texture-intensity combinative term. However, due to the correlation's dependency on the ISO speed, we postulate that: *a correlation predictor can only produce accurate predictions for images with the same ISO speed as the training images.* We call such a correlation predictor as a matching ISO correlation predictor.

To show the ISO speed's influence on correlation predictor and validate our postulate, we first compare the performance of the correlation predictors trained with (a) images with mixed ISO speeds and (b) images with the same ISO speed as the test images. We did the test on 13 cameras from the Warwick Image Forensics Dataset (An Olympus EM10 MKII camera from the dataset doesn't show strong existence of PRNU, possibly due to its image stabilization mechanism. Thus it is not included in this test). 50 flat-field images from each camera are used to extract the cameras' reference fingerprints. For each camera, we select images from three ISO speeds to form three test sets, namely ISO 100, 800, and 6400, apart from the two Panasonic LumixTZ90, which do not have ISO 6400. For these two cameras, we test on ISO 3200 images instead. Accordingly, we trained three matching ISO correlation predictors,

[0]ISO 3200 for Panasonic Lumix TZ90_1 and TZ90_2

Table 4.3: r^2 and RMSE from correlation predictions made from matching ISO and mixed ISO correlation predictors for 13 cameras in Warwick Image Forensics Dataset

	Matching ISO Correlation Predictor						Mixed ISO Correlation Predictor					
	ISO 100		ISO 800		ISO 6400[a]		ISO 100		ISO 800		ISO 6400[1]	
	r^2	RMSE	r^2	RMSE	r^2	RMSE	r^2	RMSE	r^2	RMSE	r^2	RMSE
Canon 6D	**0.7974**	**0.0194**	**0.7196**	**0.0169**	**0.5574**	**0.0116**	0.7839	0.0200	0.3983	0.0247	0	0.0292
Canon 6D MKII	**0.9518**	**0.0270**	**0.6870**	**0.0251**	**0.6912**	**0.0144**	0.9373	0.0307	0.2599	0.0386	0	0.0420
Canon 80D	**0.8593**	**0.0738**	**0.6920**	**0.0244**	**0.4108**	**0.0124**	0	0.1406	0	0.1574	0	0.0836
Canon M6	**0.8584**	**0.0182**	**0.9076**	**0.0125**	**0.7246**	**0.0083**	0.5439	0.0327	0.8042	0.0183	0.2912	0.0134
Fujifilm XA-10_1	**0.5562**	**0.0426**	**0.0582**	**0.0203**	**0.1143**	**0.0155**	0	0.0780	0.0123	0.0208	0.0809	0.0158
Fujifilm XA-10_2	**0.4648**	**0.0394**	0	0.0409	**0.1324**	**0.0151**	0.1649	0.0492	0	**0.0384**	0	0.0251
Nikon D7200	**0.7344**	**0.0145**	**0.6339**	**0.0116**	**0.4868**	**0.0101**	0.3753	0.0223	0.1737	0.0174	0	0.0170
Panasonic Lumix TZ90_1	**0.6878**	**0.0149**	0	**0.0213**	0	**0.0125**	0.2032	0.0239	0	0.0243	0	0.0208
Panasonic Lumix TZ90_2	**0.7766**	**0.0135**	**0.1448**	**0.0131**	**0.0458**	**0.0122**	0	0.0187	0	0.0140	0	0.0130
Sigma SdQuttro	**0.6758**	**0.0261**	**0.6404**	**0.0274**	**0.6361**	**0.0102**	0	0.0520	0	0.0871	0	0.0693
Sony Alpha68	**0.8614**	**0.0171**	**0.8202**	**0.0131**	**0.4684**	**0.0072**	0.7578	0.0226	0.7494	0.0155	0	0.252
Sony RX100_1	**0.4560**	**0.0446**	**0.7128**	**0.0185**	**0.7393**	**0.0151**	0	0.1021	0.5233	0.0239	0.5299	0.0203
Sony RX100_2	**0.7075**	**0.0197**	**0.6528**	**0.0168**	**0.4752**	**0.0142**	0.5713	0.0238	0.3614	0.0227	0	0.0208

[a]

Table 4.4: r^2 and RMSE for the correlation predictors generated from the matching and non-matching ISO correlation predictors for 9 cameras from Dresden Image Dataset

	Matching ISO Correlation Predictor		Non-matching ISO Correlation Predictor	
	r^2	RMSE	r^2	RMSE
Canon_Ixus55_0	**0.7012**	**0.0234**	0.6558	0.0251
Canon_Ixus70_0	**0.7111**	**0.0297**	0	0.0567
Canon_Ixus70_1	**0.7161**	**0.0267**	0.2251	0.0441
Canon_Ixus70_2	**0.6631**	**0.0306**	0	0.0664
FujiFilm_FinePixJ50_0	**0.8940**	**0.0195**	0.5130	0.0417
FujiFilm_FinePixJ50_1	**0.8928**	**0.0190**	0.8726	0.0207
FujiFilm_FinePixJ50_2	**0.9013**	**0.0199**	0.8326	0.0260
Nikon_CoolPixS710_0	**0.5400**	**0.0168**	0.3005	0.0207
Pentax_OptioA40_0	**0.3811**	**0.0315**	0	0.0596

each with 20 images of the corresponding ISO speed following the method from [23]. The correlations are computed between image blocks of 128×128 pixels. To make the comparison, for each camera, we trained another correlation predictor with 20 images randomly selected from the 60 images used for the training of the camera's three matching ISO correlation predictors. We call this correlation predictor as a mixed ISO correlation predictor. Block-wise correlation predictions are made for the test sets. For each set, we computed the coefficient of determination (r^2) and the root mean square error (RMSE) for the matching ISO and mixed ISO correlation predictors as shown in Table 4.3. We highlighted the better performance for each test set in terms of larger r^2 and smaller RMSE with bold font.

The matching ISO correlation predictors show superior performance over the mixed correlation predictors for all test sets except for the two Fujifilm XA-10 and the two Panasonic Lumix TZ90 at high ISO speeds. These two models of cameras are more prone to strong noise at high ISO speeds. As a result, the correlations with their reference PRNU become close to zero despite different image features. Due to the relatively large variance of the correlations introduced by the PRNU-irrelevant signal in the noise residuals, neither of the correlation predictors managed to produce large r^2 for the correlation predictions. However, by using the Matching ISO correlation predictor for these cameras, we notice small RMSE still can be observed. This is particularly important as the correlation predictors would not generate predictions that deviate too much from the actual correlation. False positives can be significantly reduced when we apply these correlation predictors for forgery detection.

In addition to the test on the Warwick Image Forensics Dataset, the experiments are extended to 9 cameras from the Dresden Image Dataset [116] as well. In the Dresden Image Dataset, about 150 images of natural scenes

are produced by each camera. However, as the dataset was created without considering the ISO speed as an influential factor, the images' ISO speeds span over many different values. For most ISO speeds, the number of images available is not enough for us to train a matching ISO correlation predictor using the method mentioned above and to test it with the matching ISO images. So we test the matching ISO correlation predictor on the most popular ISO speed from each camera only, each with 20 test images. For each camera, we trained a matching ISO correlation predictor with 20 images of the same ISO speed as the test images and another 20 images are selected randomly from all the images available for the training of the mixed ISO correlation predictor. r^2 and RMSE of the predictions are shown in Table 4.4. Again, the superior performance of the matching ISO correlation predictors can be observed in every case. Both the tests on images from Warwick Image Forensics Dataset and Dresden Image Dataset show that the performance of a correlation predictor may degenerate by completely ignoring the impact of ISO speed and trained images of mixed ISO speed.

Knowing that we cannot ignore the ISO speed in the correlation prediction training process, we also would like to investigate how mismatched ISO speeds of training and testing images would affect correlation prediction and subsequent forgery detection. In specific, we would like to investigate to what extent, a correlation predictor trained with images with a particular ISO speed can predict reliable correlation with images taken at other ISO speeds without significantly influencing the forgery detection results. We use Fig.4.5 to demonstrate the potential outcomes of forgery detection when the training image's ISO speed is significantly different from the test image's ISO speed.

Fig.4.5 shows the forgery detection results from tampered images with ISO speed 100, 800 and 6400 from a Canon M6. Images of the same scene taken at different ISO speeds are manipulated using Adobe Photoshop. For each image, the tampered region is replaced by using Photoshop's content-aware filling function, which leaves the tampered region at a similar noise level as its surrounding regions. We apply the Bayesian-MRF forgery detection algorithm from [24] to the images. For all the images, we set the same parameters for the forgery detection algorithm: with the interaction parameter β set to 10 and probability prior p_0 set to 0.01. The detection results show that the forgery detection algorithm works the best in terms of false detections when it is equipped with the matching ISO correlation predictor. We also notice that when we use ISO 100 correlation predictor for the forgery detection of the ISO 6400 forgery, despite the tampered region is correctly identified, there are a lot of false positives in the result. When ISO 6400 correlation predictor is used for the detection of forgery in ISO 100 forgery image, while the entire authentic region is regarded as tampered, there are parts of the tampered region still

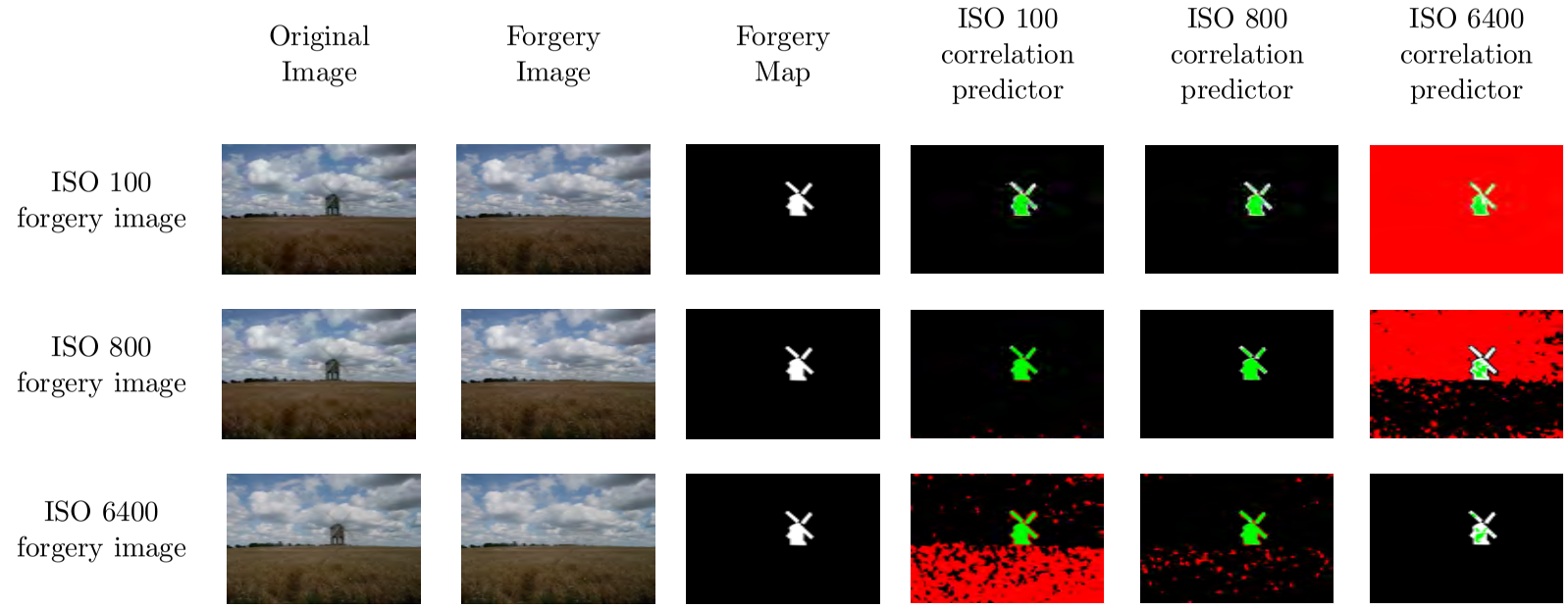


Figure 4.5: Forgery detection results on realistic forgeries from a Canon M6 with images of ISO speed 100, 800 and 6400. The images are taken with different exposure time to let them have similar exposure level. The Bayesian-MRF forgery detection algorithm is applied with the interaction parameter β set to 10 and probability prior p_0 set to 0.01. The true detections are coloured with green and red for false detections. Missed tampered pixels are shown in white.

undetected.

To explain these observations, we have to consider the two potential outcomes of using images of different ISO speeds for the training of correlation predictors: the predicted correlation being either overestimated or underestimated.

Overestimation of the correlations (when correlation predictions are larger than the actual values) often occur when we use a correlation predictor trained with images of lower ISO speeds than the test image's ISO speed. As the actual intra-class correlations will be smaller than the predicted correlation, the corresponding pixels are more likely to be labeled as tampered, which results in an increased number of false detections as we have seen in Fig.4.5. This is particularly harmful to real-life forensics. For most forgery detection algorithms, the authenticity of a pixel is checked by comparing its actual correlation with a threshold set with reference to the predicted correlations and expected inter-class correlation, which is expected to be zero. Though the actual algorithms can be different with more complexity by considering the distribution of the correlations from both inter- and intra-class as well as neighboring pixels' correlations, the comparison of whether the actual correlation sits closer to the predicted correlation or inter-class correlation when the correlation is overestimated can be a good indicator of how likely false detections can be introduced by a correlation predictor. Thus we would like to compare the two values: $d_1 = \rho - \bar{\rho}_{\text{inter}}$, which is the relative position from the inter-class correlation, $\bar{\rho}_{\text{inter}}$, to the actual computed correlation ρ and $d_2 = \bar{\rho}_{\text{intra}} - \rho$, which is the relative position of the actual correlation, ρ, to the predicted intra-class correlation, $\bar{\rho}_{\text{intra}}$. Instead of comparing the L_1 distances, we compare these two values to focus more on the situation when the correlation is overestimated, which causes the actual correlation to be a value between the expected inter-class correlation and predicted correlation. We estimate $\bar{\rho}_{\text{inter}}$ as zero and use the predicted correlation to estimate $\bar{\rho}_{\text{predict}}$, and it gives $d_1 - d_2 \approx 2\rho - \bar{\rho}_{\text{predict}}$. When $d_1 - d_2$ is negative, it indicates that the correlation has a large chance of being misidentified as an inter-class correlation.

Again, use the camera Canon M6 as an example, we show the percentages of the image blocks with $d_1 - d_2$ smaller than 0 in Fig.4.6 when we use an ISO 100 and 800 correlation predictors to predict for test images with ISO speed number of stops above the training images. The plot shows that when the test images' ISO speeds are within the one-stop range of the training images' ISO speed, there is only a relatively small portion of blocks (i.e. less than 10%) with $d_1 - d_2$ smaller than 0 for both ISO 100 and ISO 800 correlation predictors. As the deviation from the test images' ISO speed to the training images' ISO speed increases, we start to see a higher percentage from Fig.4.6, indicating an

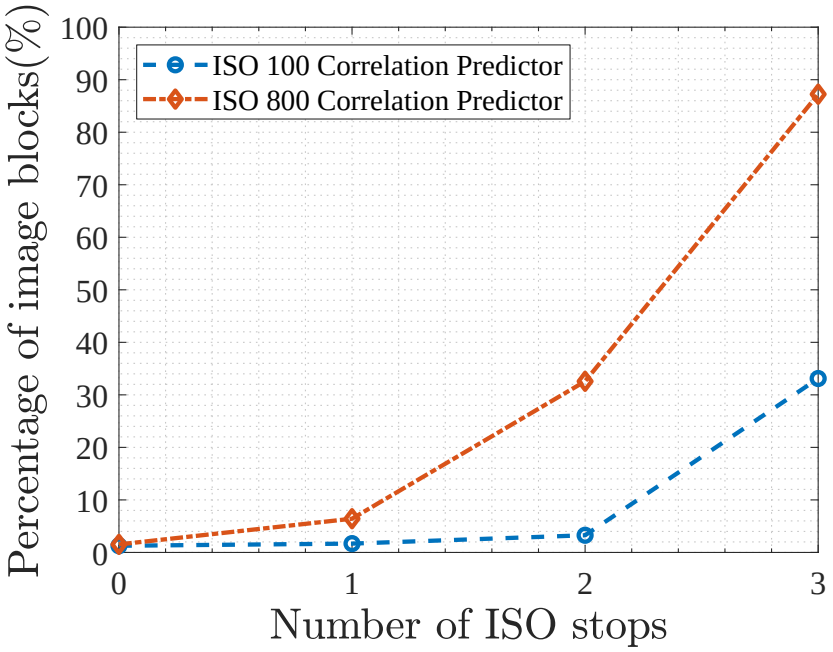


Figure 4.6: A plot of the percentages of image blocks with $d_1 - d_2$ smaller than 0 against the number of ISO stops the test image's ISO speed is above the ISO speed of the images used to train the correlation predictor for a Canon M6. The percentage indicates the portion of the authentic image blocks at risk of being misidentified as tampered blocks by forgery detection algorithms.

increased number of false detections could be introduced into forgery detection results. As we approximate $d_1 - d_2$ as $2\rho - \bar{\rho}_{\text{predict}}$, it becomes an universal problem when $\rho < \frac{1}{2}\bar{\rho}_{\text{predict}}$.

Base on the correlation model derived from Equation (4.11) and observations from experiments, we found that for image blocks of the same scene from images taken at different ISO speeds, it is generally true that the block-wise correlation in an image taken with ISO speed G_1 is twice larger than the correlation of the corresponding block from an image taken at ISO speed $G_2 = 2G_1$. Thus, we claim that $G_2 = 2G_1$ is a safe choice to be set as the largest ISO speed a correlation predictor trained with images of ISO speed G_1 can reliably predict for. Similar behavior can be observed on other cameras as well and we show the receiver operating characteristic (ROC) curve for forgery detection in Fig.4.7 for further validation.

Each ROC curve in Fig.4.7 and 4.8 is plotted by running the Bayesian-Markov random field (MRF) based forgery detection algorithm from [24] on 80 synthetic forgery images at each of the 7 presented ISO speeds. Three correlation predictors, each trained with 20 natural images taken at ISO speed 100, 800 and 6400, respectively, are used to predict the correlations for the forged images. We vary the interaction parameter β in the range of $[1, 1200]$

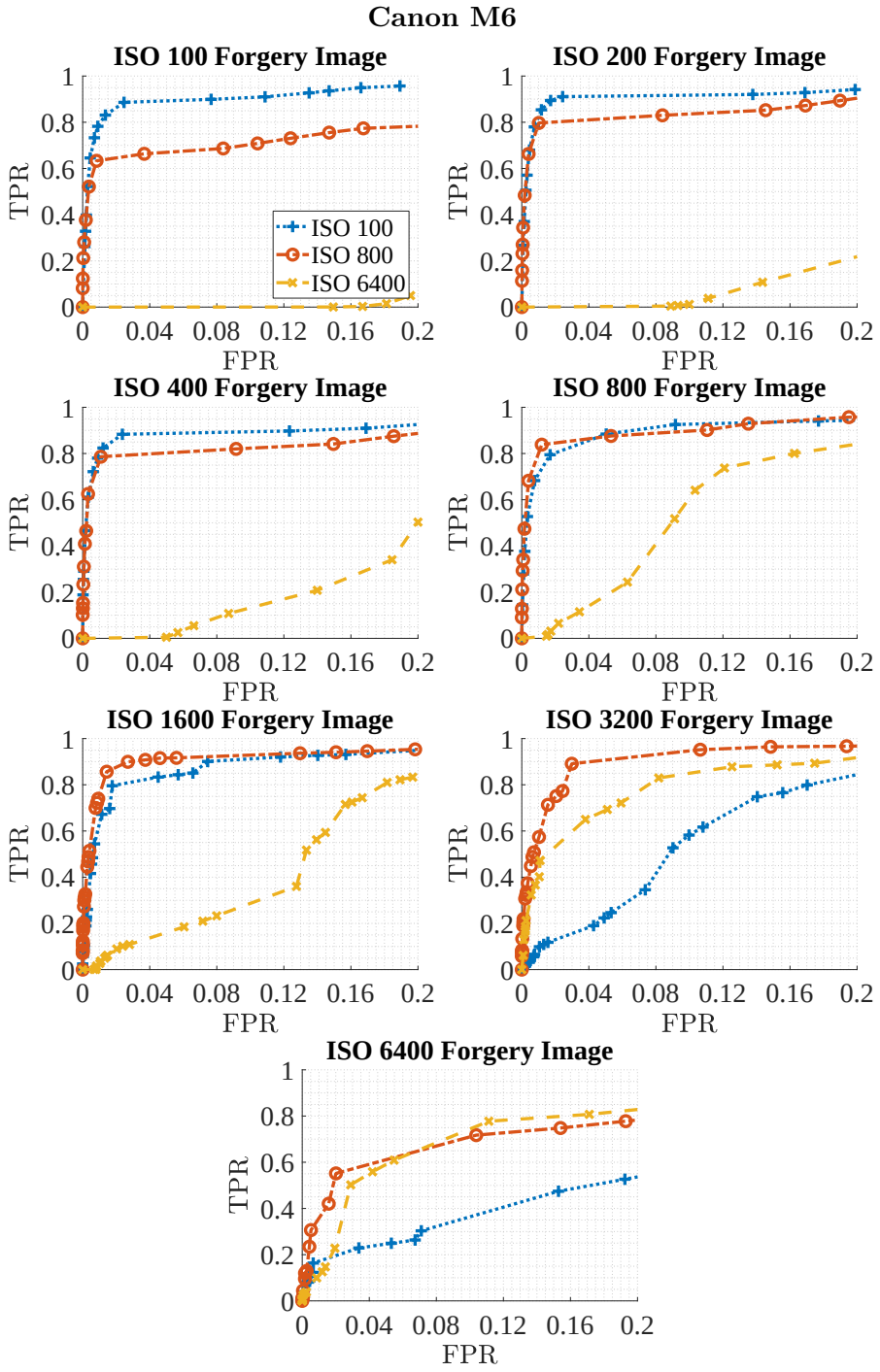


Figure 4.7: Receiver Operating Characteristic (ROC) curves of tampering localization using Bayesian-MRF forgery detection method on synthetic forgeries taken at different ISO speeds from a Canon M6. The legend shows the ISO speeds corresponding to the correlation predictors used to generate the ROC curves.

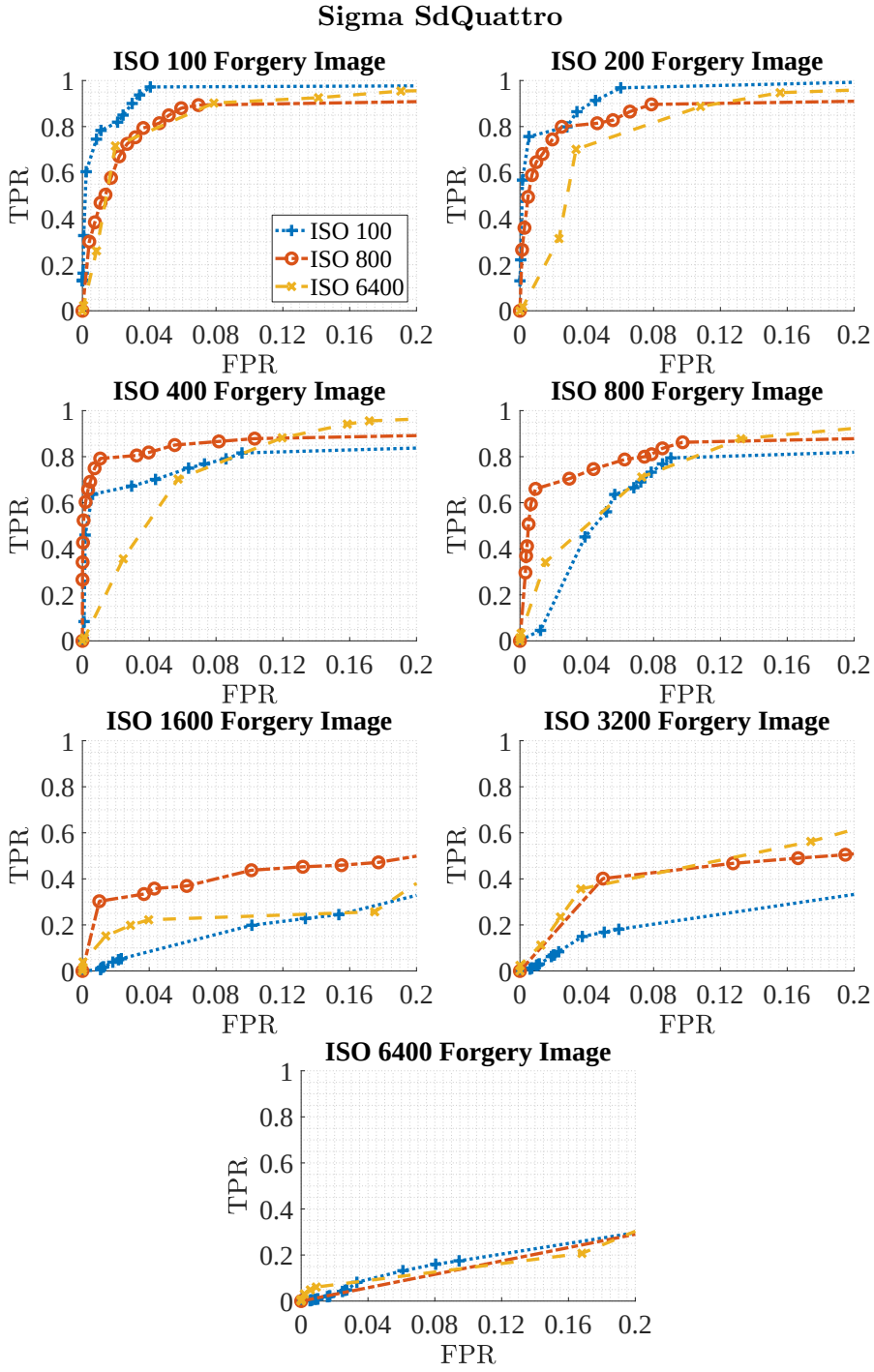


Figure 4.8: Receiver Operating Characteristic (ROC) curves of tampering localization using Bayesian-MRF forgery detection method on synthetic forgeries taken at different ISO speeds from a Sigma SdQuattro. The legend shows the ISO speeds corresponding to the correlation predictors used to generate the ROC curves.

and the probability prior p_0 between $[0, 1]$ to set different combinations of the parameters for the algorithm. This allows us to generate the enveloping curves for the ROCs to show the best performance. The 80 synthetic forged images are generated from 20 full-sized authentic images. From each full-sized image, we select 4 regions of 1024×1024 pixels. We replace the center of each 1024×1024 pixel region's center with a tampered patch of 256×256 pixels. The patch used to replace the center is cropped from the same original image but from a different position to ensure that it does not have the same PRNU. The Warwick Image Forensics Dataset provides images of the same content at different ISO speeds. This allows us to generate the synthetic forged images in the way that for one synthetic forged image at one ISO speed, we can find images of the same content at other ISO speeds as well. By doing this, Fig.4.7 and 4.8 not only allow us to compare the performance of different correlation predictors for forged images at one ISO speed but we can also systematically compare the performance of one correlation predictor for different ISO speeds.

We run the test on different cameras from Warwick Image Forensics Dataset. We show the ROC curves of two most representative cameras, a Canon M6 and a Sigma SdQuattro in Fig.4.7 and 4.8. Canon M6 represents the cameras that can generate relatively less noisy images (with a large peak to noise ratio (PSNR)) for most ISO speeds from the camera while Sigma SdQuattro represents the cameras whose image quality is highly dependent on the selected ISO speed. The false positive rate (FPR) and true positive rate (TPR) are computed at the pixel-level. As for real-life tampering localization application, we usually require the method to produce a small FPR, thus we focus on the range of $[0, 0.2]$ of FPR in the plots.

From Fig. 4.7 and 4.8, we first notice that for ISO 100, 800 and 6400 forgery images, the matching ISO correlation predictor works the best in both cameras in almost every case. The only exception is for Sigma SdQuattro ISO 6400 forgery images. In this case, despite the ISO 6400 correlation predictor can make predictions accurately as we have seen from Table 4.3, none of the three correlation predictors can produce accurate detections. This is because, for high ISO images from this camera, the images' intra-class correlations are generally very close to zero and hard to be separated from inter-class correlations. For such images, PRNU-based methods may not be the best tool to perform forgery localization. However, the ISO specific correlation predictor can still be helpful in such a scenario as it will be able to accurately predict the correlations close to zero. Thus, the users can be warned that the PRNU based methods may not be suitable under such a scenario. Overall, the results show the benefit of using a matching ISO correlation predictor for forgery detection.

For both cameras, we observe that the detection results of using the ISO 100 correlation predictors (i.e. predictors trained with images taken at ISO

speed 100) are better when the forged image's ISO speed is smaller than 400. While the Canon M6's relatively good high PSNR at higher ISO speeds allows the ISO 100 correlation predictor to perform reasonably well for a forged image with ISO speed up to 1600, it is not the case for the Sigma SdQuattro camera. From ISO 400 and above, the ISO 100 correlation predictor for the Sigma SdQuattro starts to struggle. The similar effect can be observed for ISO 800 correlation predictors when they are used to predict for images with ISO speed much higher than 800. Thus, it conforms to our argument that a predictor trained with images taken at ISO speed G_1 can perform reliably on the images taken at an ISO speed G_2 that is lower than or equal to $2G_1$. While depending on the camera, some correlation predictors may perform when the test image's ISO speed is above the range, the above argument provides a safe range for the choice of correlation predictor's training ISO speed without risking too many false detections.

Fig.4.7 and 4.8 also show the situation when the correlation predictors underestimate the test image's correlations. Underestimation often occurs when we use a correlation predictor trained with images of a much higher ISO speed than the test image's ISO speed. In the plots, we noticed that the ISO 6400 correlation predictors, especially for the Canon M6 camera, appear to have difficulty in correctly localizing the forgery for images with low ISO speed. This is because when the correlation predictor underestimates the correlations, it eventually reduces the forgery detection algorithm's capability of correctly identifying tampered pixels. Thus, to avoid the underestimation but still provide a practical range from which a training ISO speed can be conveniently selected, we empirically set the lower bound of the ISO speed a correlation predictor can be used for to half of the ISO speed of its training images. From the plots, we see by using this range, the corresponding detection results either outperform other correlation predictors or are on par with the best performance. Altogether, we conclude that for a test image taken at ISO speed G_1, using correlation predictors trained with images of ISO speed, G_2, which is in the one-stop range of G_1 ($G_2 \in [G_1/2, 2G_1]$) can produce forgery detection result without risking false detections being excessively introduced due to the correlation predictor.

4.4 ISO Specific Correlation Prediction Process

Observing the ISO speed's impact on correlation prediction, we concluded that reliable correlation predictions should be made in an ISO specific way. Thus, we propose an ISO specific correlation prediction process. To predict correlations for an image of ISO speed G_1, we have to use a correlation predictor, preferably trained with images of the same ISO speed at G_1, or similar to G_1. An ISO

speed G_2 is considered as similar to G_1 if G_2 is in the one-stop range of G_1. The images used for the training of the correlation predictor should cover diverse image feature settings: including both bright and dark scenes, highly textured and flat patterns, etc. To cover such a diverse set of image features, it usually requires a large number of images. Thus, a good correlation predictor should be trained with no less than 20 full-sized images. With a relatively large collection of images of good feature diversity taken at an ISO speed similar to the test image, the weight for each defined feature can be learned following the process presented in [23] for the correlation predictor.

In order to complete the correlation prediction process, we need to have the knowledge of the ISO speed G_1 to find images of the same or similar ISO speeds to form the training set. However, as the image in question may have undergone some unknown manipulations, either on its image content or metadata, the ISO speed information presented in the metadata can be unreliable or even unavailable. Thus, we can often face the problem when we have an image of unknown ISO speed and we would like to select images with the closest ISO speed to the image to train a correlation predictor.

As a known factor, for the same camera, the higher the ISO speed is, the higher the level of noise is introduced to the content of images. Thus, it is intuitive to infer an image's ISO speed by exploiting its noise characteristics in the content. Based on the Poissonian-Gaussian noise model [136], methods are proposed in [19, 88, 137] to infer the camera gain, g, from a RAW image, which then can be directly related to the camera's ISO speed. Despite these methods showing promising performance on RAW images, as the noise model generally cannot be applied directly to non-RAW image formats, their performance is suboptimal and cannot be practically used to infer a JPEG image's ISO speed. Furthermore, for similar reasons, though many noise level estimation algorithms [138–141] may work well on RAW images to give clues about an image's ISO speed, JPEG images still pose challenges. As JPEG is one of the most common image formats, being able to identify a JPEG image's ISO speed is a prerequisite for ISO specific correlation prediction.

Though finding an accurate noise model for a JPEG image can be of great complexity, we can simplify this problem by making the following assumption: *image patches from the same camera with similar content and JPEG quality factor should show similar noise characteristics if they are of the same ISO speed, and vice versa* as shown in Fig.4.9. Thus, we propose a method called Content-based Inference of ISO Speed (CINFISOS, pronounced as /'sin.fə.səs/) to determine an image's ISO speed by doing patch-wise noise comparison with patches of similar content from images taken with the same camera at different ISO speeds.

Consider the case when we have a query image, $\mathcal{Q}$, and t candidate training

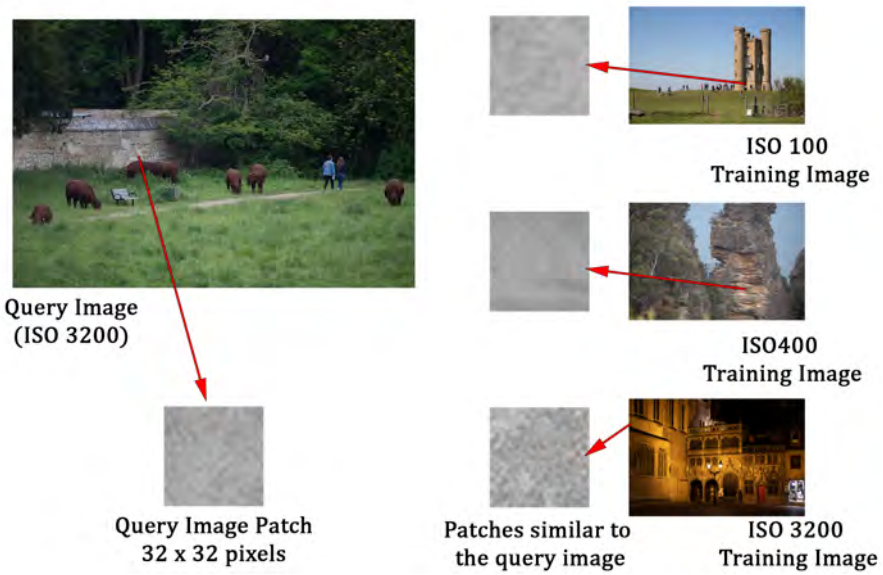


Figure 4.9: A demonstration of the idea behind the proposed ISO speed inferring method. We expect patches from different images to show similar noise characteristics if they have similar content and the same ISO speed. The example shows a patch from an ISO 3200 query image. It shows similar noise characteristics with a patch of similar content from an ISO 3200 training image.

sets, $\mathcal{S} = \{S_1, ..., S_t\}$, each consists of multiple images and the sets are with different ISO speeds. We would like to find the set with the ISO speed closest to the query image $\mathcal{Q}$. The query image is first partitioned into a set of non-overlapping patches, $\mathcal{P} = \{p_i\}$, each patch of size $d \times d$ pixels. As we would like to use the patches to best represent the image's noise characteristics, patches with too many dark and saturated pixels in any color channel should be removed. We consider the patches in the RGB color space. For each pixel q in the jth channel of the patch, p_i^j, the pixel is considered as dark or saturated if its pixel value $\boldsymbol{I}(q)$ is not in the range $[\lambda_1, \lambda_2]$:

$$\boldsymbol{U}(q) = \begin{cases} 1, \text{if } \boldsymbol{I}(q) < \lambda_1 \text{ or } \boldsymbol{I}(q) > \lambda_2 \\ 0, \text{otherwise} \end{cases} \tag{4.14}$$

To form a set $\hat{\mathcal{P}}$, which does not contain dark or saturated patches, the ith patch is removed from $\mathcal{P}$ if $\forall j(\sum_{q \in p_i^j} \boldsymbol{U}(q) > \lambda_\tau d^2)$, when the ratio of the dark or saturated pixels in every channel of the patch is over a limit λ_τ. In addition to removing the dark and saturated pixels, the image's noise characteristics can be better revealed by including only the less textured patches. Thus, we only keep m least textured patches in $\mathcal{P}_\mathcal{Q}$, the set of patches that we believe can best represent the query image's noise characteristics. To evaluate how

textured a patch is, we use the texture feature definition from [23] but extends its definition to patches of three color channels by a simple summation:

$$f_T(p_i) = \sum_{j=1}^{3}(\frac{1}{d^2}\sum_{q\in p_i^j}\frac{1}{1+\text{var}_5(\mathbf{F}(q))}) \tag{4.15}$$

where $\mathbf{F}()$ is the high-pass filter and $\text{var}_5()$ measures the variance of 5×5 neighbourhood. The feature f_T is defined in the range $[0,1]$ with lower values for more textured patches. We select m least textured patches from $\hat{\mathcal{P}}$ to form the set of qualified query image patches $\mathcal{P}_\mathcal{Q}$:

$$\mathcal{P}_\mathcal{Q} = \{p_i | (p_i \in \hat{\mathcal{P}}) \wedge (f_T(p_i) > f_{Tm+1})\} \tag{4.16}$$

f_{Tm+1} is the texture feature of the $m+1$th least textured patch from $\hat{\mathcal{P}}$. As $\mathcal{P}_\mathcal{Q}$ only contains patches with relatively smooth texture, we can approximate their image content by applying a low pass filter. We implement the method of finding patches with similar content using a block-matching method similar to [129]. The distance between two patches in each color channel is measured as the Euclidean distance between the discrete cosine transforms (DCT) of the two with hard thresholding applied. The overall distance between two patches is the summation of the distances in the three color channels:

$$\Delta(p_i, p_k) = \sum_{j=1}^{3}\|\Gamma(\text{DCT}(p_i^j), \lambda_{\text{DCT}}) - \Gamma(\text{DCT}(p_k^j), \lambda_{\text{DCT}})\|_2 \tag{4.17}$$

where $\Gamma(x, \lambda_{\text{DCT}})$ is the hard thresholding operation:

$$\Gamma(x, \lambda_{\text{DCT}}) = \begin{cases} x, \text{if } x > \lambda_{\text{DCT}}, \\ 0, \text{otherwise} \end{cases} \tag{4.18}$$

For each patch p_i in $\mathcal{P}_\mathcal{Q}$, from each candidate training set S_k, n patches with the least distance to p_i will be selected. Though the exhaustive search for the patches with the shortest distance is computationally expensive, this step can be easily parallelised. We call this set of selected patches as $\mathcal{P}_k^i$. We define the distance, which measures the sum of the absolute differences in noise characteristics in all three color channels from each patch p_i in $\mathcal{P}_\mathcal{Q}$ to each candidate training image set S_k, as:

$$D(p_i, S_k) = \sum_{j=1}^{3}(|\text{var}(p_i^j - \tilde{p}_i^j) - \frac{1}{n}\sum_{p_l\in\mathcal{P}_k^i}\text{var}(p_l^j - \tilde{p}_l^j)|) \tag{4.19}$$

where $\tilde{p}_l^j$ is the low-pass filtered version of the patch p_l of the jth channel:

$$\tilde{p}_l^j = \text{IDCT}(\Gamma(\text{DCT}(p_l^j), \lambda_{\text{DCT}})) \tag{4.20}$$

For each patch p_i in $\mathcal{P}_{\mathcal{Q}}$, it will have a vote for a candidate training set, S_k, who has the smallest $D(p_i, S_k)$. The candidate training set with the closest ISO speed to the query image will be determined by a simple majority vote from all the patches in $\mathcal{P}_{\mathcal{Q}}$. The ISO speed that receives the majority votes will be deemed as the ISO speed of the query image and the correlation predictor can be trained with the corresponding images.

4.5 Experiments

4.5.1 Inferring ISO Speed with CINFISOS

To test the performance of the proposed CINFISOS, we conduct experiments on our Warwick Image Forensics Dataset. In the previous section, we concluded that for a correlation predictor trained with ISO speed G_1, reliable correlation predictions can be made for images taken with ISO speed in the range of $[G_1/2, 2G_1]$. Therefore, to select a correlation predictor trained with images of an ISO speed suitable for the image in question, the inferred ISO speed only needs to be within the one-stop range of the real value. As a result, we only need a few candidate training sets, S_k, to cover a broad range of ISO speeds to give reliable correlation predictions.

In our experiments, for each camera in the Warwick Image Forensics Dataset, we have three candidate training sets with images of ISO speed 100, 800 and 6400, respectively (with the exception for the two Panasonic Lumix TZ90, of which we select the ISO 3200 candidate training set instead of the ISO 6400 training set). These three ISO speeds are selected as they cover a broad range of commonly used ISO speeds. Besides, we deliberately avoid overlapping between the one-stop range of the ISO speeds, each of the three candidate ISO speed can predict for, to make it easier for the performance evaluation.

To apply CINFISOS, we set the following parameters. The size of each query image patch is 32×32 pixels. $m = 50$ is the number of patches in the qualified query set $\mathcal{P}_{\mathcal{Q}}$. λ_{DCT} is set to 13.0315 in a similar manner as how it is set in [129]. For each query patch, we find 5 similar patches from each candidate set. For each camera in the Warwick Image Forensics Dataset apart from the two Panasonic Lumix TZ90, we have 20 query images, each with ISO speed 100, 200, 400, 800, 1600, 3200 and 6400 in the JPEG format. Each candidate training set consists of 20 images. For the two Panasonic Lumix TZ90, in addition to the fact that ISO 6400 images are unavailable, we also excluded ISO 1600 query images as both ISO 800 and 3200 can be considered

Table 4.5: Patch level accuracy of the proposed ISO speed inferring method on images from Warwick Image Forensics Dataset

	ISO 100	ISO 200	ISO 400	ISO 800	ISO 1600	ISO 3200	ISO 6400
Canon 6D	0.954	0.843	0.619	0.740	0.637	0.755	0.806
Canon 6D MKII	0.999	0.952	0.593	0.795	0.764	0.723	0.744
Canon 80D	0.990	0.893	0.789	0.882	0.851	0.879	0.997
Canon M6	1.000	0.937	0.682	0.869	0.836	0.911	0.983
Fujifilm XA_10_1	0.725	0.574	0.543	0.666	0.612	0.704	0.668
Fujifilm XA_10_2	0.699	0.602	0.587	0.673	0.578	0.625	0.654
Nikon D7200	0.998	0.891	0.734	0.859	0.800	0.860	0.918
Olympus EM10 MKII	0.989	0.928	0.631	0.694	0.712	0.697	0.731
Panasonic Lumix TZ90_1	0.961	0.802	0.554	0.581	N.A.	0.720	N.A.
Panasonic Lumix TZ90_2	0.908	0.769	0.580	0.576	N.A.	0.708	N.A.
Sigma SdQuttro	0.881	0.825	0.512	0.716	0.565	0.601	0.642
Sony Alpha68	0.948	0.883	0.714	0.850	0.748	0.863	0.993
Sony RX100_1	0.913	0.856	0.741	0.869	0.677	0.549	0.648
Sony RX100_2	0.998	0.915	0.791	0.837	0.625	0.610	0.763

as inferred correctly.

We run the experiment with a desktop equipped with an Intel Core i7-9700K CPU. With the afore-mentioned setup, it takes around 130 seconds for CINFISOS to run on a full-resolution query image (e.g. 4160×6240 pixels for an image from a Canon 6D MKII), including the exhaustive search for similar patches among 60 full-resolution training images. The patch-level accuracy, which measures the percentage of patches voting correctly for the inferred ISO speed, is reported in Table 4.5. We notice that the accuracy varies greatly between cameras at different ISO speeds but the accuracy is above 0.5 in every case. It means that overall, every single patch is more likely to vote correctly. Given this patch-level accuracy, a 99.52% accuracy at the image-level is observed with only 9 out of 1880 test images wrongly inferred.

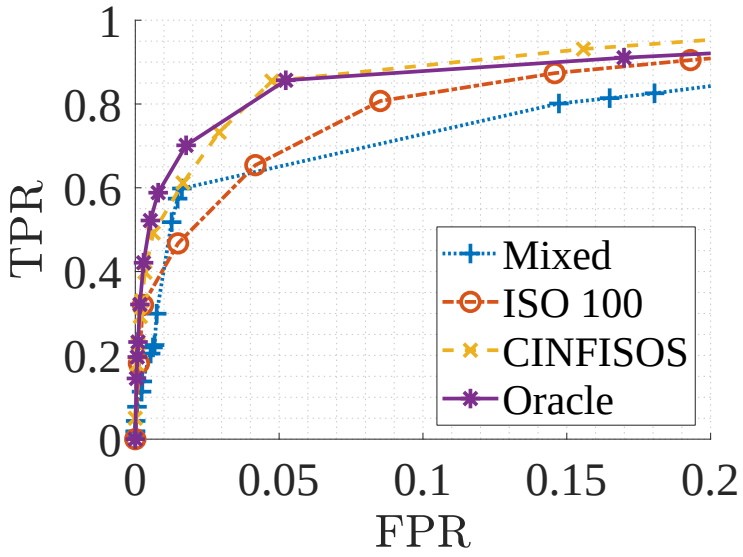
4.5.2 Forgery Detection with ISO Specific Correlation Prediction

The high accuracy of CINFISOS in identifying the ISO speed of an image within its one-stop range allows us to conduct the proposed ISO specific correlation prediction process even when we do not know the test image's ISO speed. Thus, we would like to test the performance of the proposed ISO specific correlation prediction process in terms of forgery detection.

We apply the Bayesian-MRF forgery detection algorithm[24] on the synthetic forgery images from two cameras: a Canon M6 and a Sigma SdQuattro for the test. The images are the same as the ones used in Section 4.3. There are 560 synthetic images from each camera and they are equally distributed over 7 different ISO speeds (namely ISO speed 100, 200, 400, 800, 1600, 3200 and 6400). We carry out the proposed ISO specific correlation prediction process in two ways: (a) using the proposed CINFISOS to determine whether a correlation predictor is suitable for the test image, and (b) with an oracle correlation predictor. With the aforementioned one-stop range setting, we only need three correlation predictors, namely an ISO 100, an ISO 800 and an ISO 6400 correlation predictor to cover the whole range of the ISO speeds we need to predict for with CINFISOS. We apply CINFISOS on each synthetic image to determine which of the three correlation predictors should be used to produce the predictions of each image. The oracle correlation predictor uses a matching-ISO correlation predictor for each image according to its ISO speed information. We trained 7 different correlation predictors for the 7 different ISO speeds presented in this test, each with 20 natural images, to realise the oracle correlation predictor.

We compare the forgery detection results by our proposed ISO specific correlation prediction process against the results by using correlation predictions

(a) Canon M6



(b) Sigma SdQuattro

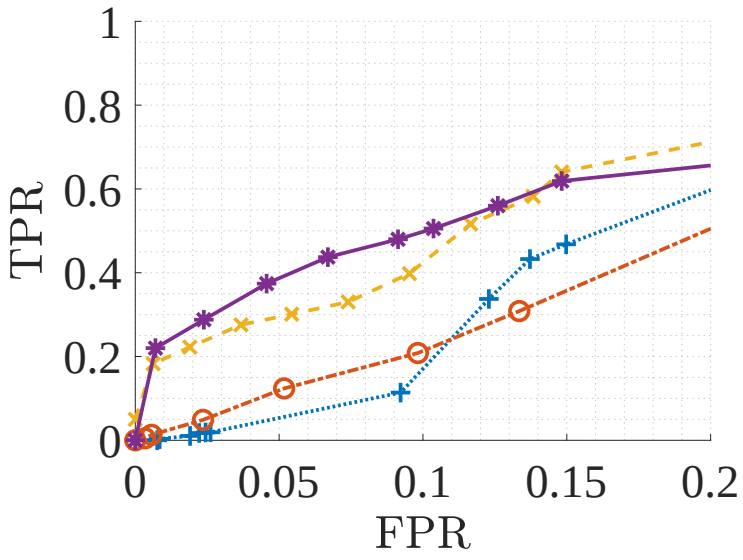


Figure 4.10: The ROC curves depicting the performance of detector with various correlation predictors tested on 560 synthetic forgery images of 7 different ISO speeds for two cameras (a) a Canon M6 and (b) a Sigma SdQuattro. Forgery detections are carried out with the Bayesian-MRF forgery detection algorithm with correlation predictions generated from (i) a mixed ISO correlation predictor (ii) an ISO 100 correlation predictor (iii) the proposed ISO specific correlation prediction process with CINFISOS and (iv) the proposed ISO specific correlation prediction process with an oracle correlation predictor.

with a mixed ISO correlation predictor and an ISO 100 correlation predictor. Mixed ISO correlation predictors represent the situation when we select training images randomly without considering the images' ISO speeds. Thus, the mixed ISO correlation predictors' performance can be viewed as the baseline for the forgery detection results when we disregard the impact from ISO speed on correlation prediction completely. For each camera, the mixed ISO correlation predictor is trained with 20 training images randomly selected from the 60 images of three different ISO speeds. The ISO 100 correlation predictor is the same as the one used in our proposed ISO specific correlation prediction process. We vary the interaction parameter β and the probability prior p_0 for the Bayesian-MRF forgery detection method to generate the enveloping ROC curves. Each data point on the curve is generated by summing the detection results of the 560 synthetic images from each camera. The ROC curves for the detection results are shown in Fig.4.10. We focus on the low false positive rate range of $[0, 0.2]$.

Unsurprisingly, the detection result from the oracle correlation predictor comes as the best above all the predictors for both cameras. However, the detection results based on the proposed CINFISOS are comparable to the oracle correlation predictor's ones. It shows the effectiveness of the proposed CINFISOS and validates that the one-stop range for ISO speed prediction is a feasible choice without significantly sacrificing the forgery detection performance. In comparison, the mixed ISO and ISO 100 correlation predictors have worse performance. Though in Fig.4.7, we have noticed that the ISO 100 correlation predictor can predict well for images with ISO speed up to 1600, its poor performance on images of higher ISO speed is evident. Thus, it is not a good choice to use a correlation predictor trained with low ISO speed for all the images. To conclude, the proposed ISO specific correlation prediction process shows superior performance in terms of forgery detection.

4.6 Conclusion

In this chapter, we did both analytical and empirical studies on the impact of different camera sensitivity (ISO speed) settings on PRNU-based digital forensics. First, we show how the correlation between an image's noise residual with the device's reference PRNU can be dependent on the image's ISO speed. With this dependency in mind, we empirically show how mismatched ISO speeds may influence the correlation prediction process. Thus, we proposed an ISO-specific correlation prediction process to be used in PRNU-based forgery detection. To address the problem that the information about the ISO speed of an image may not be available, a method called Content-based Inference of ISO Speed (CINFISOS) is proposed to infer the image's ISO speed from

its content. Clear improvements are observed in correlation predictions and forgery detection results by applying our proposed ISO specific correlation prediction process with CINFISOS. By pointing out the influence of camera sensitivity setting on PRNU-based forensic methods, the provided solutions from this chapter can make the forensic analysis more reliable and trustworthy.

Chapter 5

PRNU-based Provenance Analysis for Instagram Photos

In Section 2.3 of Chapter 2, we discussed how PRNU-based source-oriented image clustering can be performed using the similarity measurements between PRNUs extracted from images. Most PRNU-based source-oriented image clustering methods assume that the extracted PRNUs are pristine such that there are no external factors that may alter the similarity measurements. However, this might not be true for images from social networking sites, e.g. Instagram, Facebook, *etc.* These image-sharing sites usually provide users with some image editing tools at the image uploading stage, for example, the image filters used by Instagram. Thus, the users may apply common image editions on the uploaded images. These common image manipulations may accidentally increase the similarity measurements between the extracted PRNU and exert influence on the PRNU-based source-oriented image clustering methods. Thus, in this chapter, using the image filters from Instagram as an example, we investigate these common image editing tools' impact. Realizing that the existing PRNU-based source oriented clustering methods may fail completely on Instagram images due to these common image editing tools, we propose a novel three-step clustering framework to perform source-oriented clustering on Instagram images.

The rest of this chapter is organized as follows. An introduction to the background is given in Section 5.1. Section 5.2 shows the preliminary test of the existing PRNU-based source camera identification and clustering methods on images from Instagram. The proposed three-step clustering method is shown in Section 5.3. Section 5.4 presents the experimental results while Section 5.5 draws the conclusion.

5.1 Introduction

With the rapid development of mobile networks and the ever-increasing prevalence of smartphones, photo-sharing social networking sites (SNSs), such as Instagram, Facebook and Flickr, have become ubiquitous in our daily life. With millions of daily active users, these SNSs not only provide effective platforms for information sharing but also exert huge influence on commerce and politics. However, due to the convenient and broad reach of these platforms, they have been increasingly exploited for various malicious purposes, e.g. fraudulent advertisement, fictitious news or even terrorism. Meanwhile, the sheer volume of user-generated content on these platforms provides a rich source of evidence acquisition for forensic investigations. Thus, in recent years there have been growing interests in developing forensic tools and techniques to facilitate the investigations on the data collected from SNSs. One important related topic is the provenance analysis of images from SNSs. The provenance information of digital images is essential for forensic investigations. For example, when a forensic investigator is dealing with a set of images of unknown sources from multiple social network accounts, revealing the source devices of the images can help the investigator to focus on the images from the same source. In addition, linked and fake social network accounts can be discovered by finding images from the same source device across different accounts. This is because different accounts with photos taken with the camera are likely to be closely linked (e.g., between family members or friends) or fake accounts used in sybil attacks. With these telltale provenance information, more effective investigations can then be carried out. Though occasionally, one may use the metadata of an image to retrieve its provenance information, these information can still be questionable as the metadata could be edited easily. Moreover, many SNSs deliberately delete metadata from the images when they are uploaded. The unavailability of the provenance information may entail content-based analyses when rigorous forensic investigations are required.

Among various techniques used for analyzing images' provenance, the PRNU-based methods have drawn extensive attention from researchers. PRNU has been proved to be a powerful tool for provenance analysis, such as source camera identification (SCI) [22, 130, 142], source-oriented clustering (SOC) [25, 26, 90, 92, 95, 97, 98, 101]. Generally speaking, SCI is a relatively easy task provided that the high-quality reference PRNUs are available. In comparison, it is more challenging for SOC, where we aim to group a set of images of unknown sources into a number of clusters, such that the images in the same cluster are taken by the same camera. For this task, we often face the challenges of an unknown number of source devices and low-quality of the PRNUs extracted from single images. Many techniques or combinations of them have been proposed

for PRNU-based SOC, including the methods based on hierarchical clustering [90, 91], graph-based approaches [26, 92, 95, 97], constraint optimization [98] and Markov random field [25] as reviewed in Chapter 2. However, due to the unavailability of the reference PRNUs, these algorithms have to rely on the pairwise correlations between individual noise residuals, which are more susceptible to PRNU-irrelevant interferences, especially for the images from SNSs that may have undergone a series of post-processing operations. This raises doubts about whether PRNU-based provenance analysis methods remain effective on images from social network sites.

Goljan *et al.* [143] perform a large-scale test of PRNU-based camera identification on images downloaded from Flicker and show very promising results with a small false rejection rate <0.0238 at a false acceptance rate $<2.4\times10^{-5}$ for 6896 cameras with 150 different camera models. However, comparing to other social networking platforms, Flickr allows the uploaded images to be stored in their original resolution with no or very little compression, so it does not fully reflect the difficulty of the problem we usually face when performing image provenance analysis on other SNSs. Satta and Stirparo [144] use PRNUs to build the link between a photo and the user accounts of the person that has shot the photo. A probe photo is considered to be from the account containing the image with the highest matching score to the probe photo. Their method achieves a recognition rate of $\sim 50\%$ by evaluating 2896 images from 30 different accounts across different SNSs, namely Flickr, Facebook, Google+ and personal blogs. The low recognition rate and the lack of in-depth investigation into the effect of image operations make it necessary to conduct further studies on the PRNU-based provenance analysis of images from SNSs.

More recent work [145, 146] discover that different SNSs may apply different image manipulations, which leave distinctive artifacts that can be used to trace the origin SNSs of the images. Moreover, they show how common it is for the SNSs to apply 'hidden' image manipulations, such as resizing and re-compression, to fulfill the system requirement, which may affect the PRNU and pose challenges to PRNU-based provenance analysis. Apart from the above-mentioned image manipulations, many SNSs also provide explicit image manipulation tools to allow the users to edit image effects according to their own preferences, with the 'Filters' from Instagram being the most famous example. While these tools enrich the user experience, they may also manipulate the images in a way that may make the PRNU-based provenance analysis method ineffective. In this chapter, we will investigate the effects of Instagram filters and propose a new method to mitigate the impact of image filtering.

To carry out the investigation, we prepared a dataset $\mathscr{D}$ with a large number of images of known sources and applied different image filters to them. We

Table 5.1: An overview of different datasets used for different parts of the work with information including the source of the original images. $\mathscr{D}$, which is derived from VISION dataset, is the main dataset used in this chapter, including the training and testing of the proposed CNN-based filter classifier in Section 5.4.1. $\mathscr{D}_{\text{SCI}}$ is a subset of $\mathscr{D}$, which is used to test device fingerprint based SCI in Section 5.2.1. $\mathscr{D}_{\text{Dresden}}$ is derived from Dresden Image Database and used to show the proposed CNN-based filter classifier is not overfitted to the training cameras in Section 5.4.1. $\mathscr{D}_2$, $\mathscr{D}_3$, $\mathscr{D}_4$, $\mathscr{D}_5$, $\mathscr{D}_6$ are subsets of $\mathscr{D}$ with different sizes, used to test proposed clustering framework in Section 5.4.3

Dataset	Source	No. of Devices	No. of Images		
			total	per device	per filter
$\mathscr{D}$	VISION	25	96,660	> 2466	5370
$\mathscr{D}_{\text{SCI}}$	VISION	25	22,500	900	1250
$\mathscr{D}_{\text{Dresden}}$	Dresden	11	29,700	2700	1650
$\mathscr{D}_2$	VISION	25	900	36	50
$\mathscr{D}_3$	VISION	25	1,350	54	75
$\mathscr{D}_4$	VISION	25	1,800	72	100
$\mathscr{D}_5$	VISION	25	2,250	90	125
$\mathscr{D}_6$	VISION	25	2,700	108	150

selected $5,370$ images captured by 25 cameras, with at least 137 images from each camera, from the VISION image dataset [117]. The images are aligned to the same horizontal orientation according to their EXIF data and cropped to the size of $1080 \times 1080 \times 3$ pixels to match the default image size on Instagram. For each image, we applied 17 different Instagram image filters by running the Instagram application on an iOS simulator. Thus, together with the original version, we generated 18 different versions of each image and in total $96,660$ images for the use in our work. Fig. 5.1 shows a sample image for each filter together with the original image (labelled as 'Normal' filter as it is termed on Instagram). In addition, we also processed images from Dresden Image Dataset [116] and form various subsets of $\mathscr{D}$ to carry out tests on different aspects of device fingerprint based provenance analysis and our proposed framework. An overview of these datasets are shown in Table 5.1.

5.2 Existing PRNU-based Provenance Analysis on Instagram Images

5.2.1 PRNU-based SCI for Instagram Images

In this section, we investigate the effect of different Instagram image filters on the task of PRNU-based SCI. Specifically, we perform SCI by examining

Figure 5.1: Example images of the 17 Instagram filters together with the original image (Normal) used in our experiment.

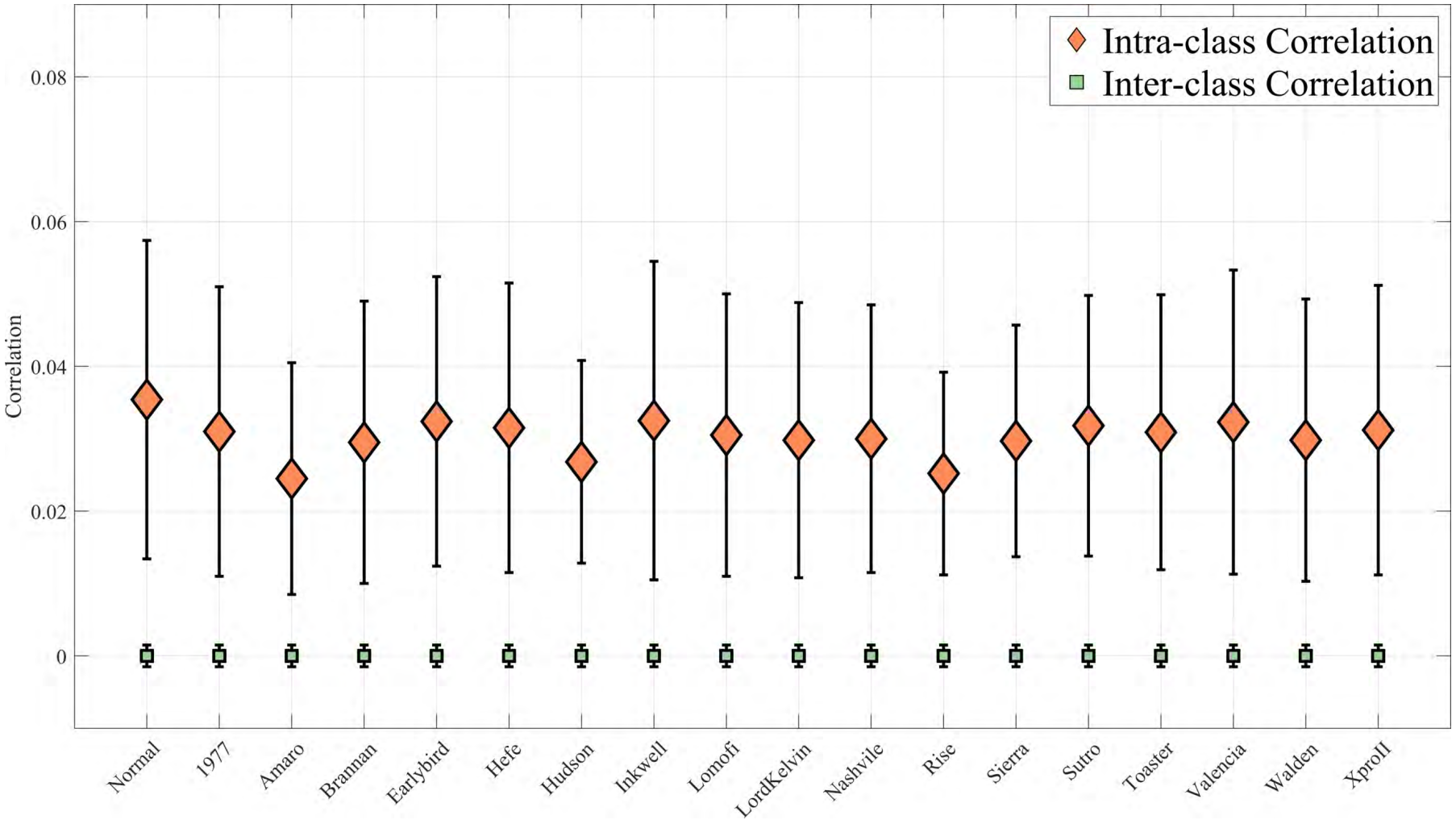


Figure 5.2: The correlation distributions for filtered images from an iPhone4s with its reference PRNU with the central points representing the means and the error bars for the standard deviations. The distributions of the correlations between filtered inter-class images with the smartphone's original reference PRNU are also shown in the figure.

Table 5.2: Source camera identification accuracy (Acc.) for different Instagram image filters

Filter	Normal	1977	Amaro	Brannan	Earlybird	Hefe
Acc. (%)	95.52	95.04	95.04	95.04	95.28	95.12
Filter	Hudson	Inkwell	Lomofi	LordKelvin	Nashville	Rise
Acc. (%)	95.04	95.20	95.12	95.20	95.04	95.04
Filter	Sierra	Sutro	Toaster	Valencia	Walden	XproII
Acc. (%)	95.20	95.44	95.36	95.52	95.20	95.28

the correlations between the noise residuals extracted from the *filtered* images with the reference PRNUs, each of which is estimated from 50 flat-field images taken by the same camera. Note that these flat-field images are original images to ensure the high quality of reference PRNUs. Thus, the performance of the source camera identification task can serve as the baseline for the quality of the PRNU embedded in the Instagram images. BM3D de-noising algorithm [129] is used to extract the noise residual for each image. For the reference PRNU of each camera, its correlations with $21,150$ inter-class *original* images (i.e. the ones from different source cameras) are computed to estimate the inter-class correlation distribution. After the inter-class distribution is estimated, we determine a decision threshold $\{\tau_i\}_{i=1}^{25}$ for each camera according to the corresponding inter-class correlation distribution based on the Neyman-Pearson criterion (by setting the false positive rate as 1×10^{-3}). We formed a testing dataset $\mathscr{D}_{\text{SCI}}$ with 50 test images $\{\boldsymbol{I}_l^{ij}\}_{l=1}^{50}$ for each camera i processed by each filter j randomly selected from $\mathscr{D}$. For each test image $\boldsymbol{I}_l^{ij}$, the largest correlation $\rho_{i^\star}$ among the correlations $\{\rho_i\}_{i=1}^{25}$ between its noise residual $\boldsymbol{n}_l^{ij}$ and the reference PRNUs $\{\boldsymbol{r}_i\}_{i=1}^{25}$ of candidate cameras is compared with the pre-defined threshold τ_{i^*} to examine whether the image is from the camera i^* or from an unknown source. The accuracy of the SCI for each filter is shown in Table 5.2. In addition, as all devices show similar behaviour, to explicitly demonstrate the quality of PRNU embedded in the filtered images, we select one camera (an iPhone4s) as an example and plot the intra-class correlation distributions between the test images with the reference PRNU for different filters in Fig. 5.2, where we use central points and error bars to represent the means and standard deviations of the distributions, respectively.

Table 5.2 shows that for each filter, the identification accuracy for the images processed by the same filter is comparable to that for the 'Normal' images. This is no surprise when we look at the correlation distribution plot in Fig. 5.2. As different devices show similar behaviour, we use an iPhone4s as an example. First, we notice that different Instagram image filters have almost no impact on the inter-class distribution. Secondly, when we compare

the intra-correlation distributions from different image filters to the original images ('Normal'), we can only notice small reductions in intra-class correlation values and such reductions are insignificant compared to the difference between intra- and inter-class distributions. This explains why SCI remains accurate when image filters are applied to the images. Most importantly, these results imply that *the PRNU is well preserved in the filtered images though it may be affected differently by filtering operations*. In other words, PRNU is still useful for image provenance analysis even after the Instagram image filters have been applied.

5.2.2 PRNU-based SOC for Instagram Images

While the above SCI results show that the PRNU is preserved in the filtered images, SOC relying on the pairwise similarities between the noise residuals of individual images can be more challenging. For SCI, as the reference PRNU is immune from the filter-related artifacts, the inter-class correlation is unlikely to be altered. However, for SOC, the common artifacts introduced by the same filter may falsely increase the pairwise correlations between inter-class images, which might lead to *filter-oriented* rather than *source-oriented* clustering results. Thus, SOC is more vulnerable to these filter-related artifacts.

To investigate further, we test the images with the fast clustering (FC) method from [25], which has shown good precision and recall rates when applied on unedited original images from public image datasets. As a whole, we perform the SOC task on a test dataset, namely $\mathscr{D}_4$, which consists of 1800 images with 72 images from each of the 25 cameras in $\mathscr{D}$. The 72 images of each camera consist of 4 images randomly selected from those filtered by each of the 18 filters, which results in $4\times25=100$ filtered images for each filter. As shown in Table 5.3, the precision, recall and F1-measure are 61.11%, 39.17% and 47.74%, respectively, which are much lower even than the results (precision: 92.1%, recall: 81.2%, F1-measure: 86.3%) reported for the hard dataset $\mathcal{D}_4$ in [25]. To show that the performance is not biased to a specific algorithm, results are also shown in Table 5.3 for the hierarchical clustering (HC) method [90], the normalized cut-based clustering (NCUT) method [95] and consensus correlation clustering (CCC) method [97]. These methods show good performance on the task of clustering 1000 'Normal' images from 25 different devices, with F1-measures of 83.69%, 85.30% and 86.75% for HC, NCUT and CCC, respectively. This shows that these PRNU-based clustering methods are effective for Instagram images when the filters are not applied. However, for the task of clustering images with different filters applied, the low F1-measure rates in Table 5.3 for all the algorithms clearly show that it is a common challenge for existing PRNU-based SOC algorithms to analyze

Instagram images.

Additionally, to investigate how each filter affects the PRNU-based clustering method, we also perform separate clustering on the images filtered by the same filter. Because the fast clustering method from [25] shows the best performance among the 4 methods shown in Table 5.3, which means it is least affected by the Instagram filters, we use it as an example for this investigation. For each filter, we select 40 images from each of the 25 cameras in $\mathscr{D}$. Thus, for this experiment, the clustering for each filter is evaluated on 1000 images. The results are shown in Table 5.4. An interesting observation made from Table 5.4 is that, among the filters we have tested, some filters dramatically deteriorate the clustering performance while the others result in comparable clustering performance to that on *original* images. We, therefore, refer to the former set of filters as Group M because the filters are malignant for PRNU-based SOC and the latter set of filters as Group B because the filters are 'benign'. When there is no ambiguity, we will also use Group M and Group B to refer to the images filtered by the former and the latter set of filters, respectively. We find that for Group M filters, the images are clustered into a single cluster, which is responsible for the low precision rate of 4.0%, i.e. each of the 25 camera accounts for 40 images in the resultant single cluster. This can be largely attributed to the common artifacts shared between the images filtered by the same filter. An example is shown for filter 'Hefe' in Fig. 5.3(a), where we plot the intra- and inter-class correlation distributions for original images (i.e. 'Normal') and the images processed by filter 'Hefe'. We also show the two corresponding grayscale plots of the 1000×1000 pairwise correlation matrices computed with 1000 'Normal' and 'Hefe' images, respectively, in Fig. 5.3(b). We can see that there is an apparent increase in mean and variance for both inter- and intra-class distributions for filter 'Hefe'. It is noteworthy that the increase of intra-class correlations is caused by the filter-related artifacts, thus it is not beneficial for *camera-oriented* clustering but rather gives rise to misleading *filter-oriented* results. Such an increase in inter-class correlations can be observed for all Group M filters. Therefore, a clustering algorithm that can mitigate the effect of the artifacts introduced by the filters in Group M is needed for the effective provenance analysis of Instagram images.

5.3 Proposed Method

In the previous section, we have demonstrated the difficulty in PRNU-based SOC on Instagram images, which arises mainly because of the artifacts introduced by the filters in Group M. Inspired by the success of the PRNU-based SCI, for which the reference PRNUs are available, we develop a framework that first performs clustering on the images in Group B and use the resultant

Table 5.3: Clustering result on 1800 images with mixed filters and native images using the fast clustering (FC) method, the hierarchical clustering (HC) based method, the normalized cut-based clustering (NCUT) based method and the consensused correlation clustering (CCC) based method.

%	Precision	Recall	F1-measure
FC	61.11	39.17	47.74
HC	56.06	37.88	45.21
NCUT	5.61	46.76	10.02
CCC	98.95	2.74	5.33

Table 5.4: SOC results for different Instagram Image filters using the fast clustering method from [25]. The filters in Group M are highlighted with gray background.

%	Precision	Recall	F1-measure
Normal	94.70	78.92	86.09
1977	93.90	78.25	85.36
Amaro	4.00	100	7.69
Brannan	95.30	79.42	86.64
Earlybird	93.60	80.69	86.67
Hefe	4.00	100	7.69
Hudson	4.00	100	7.69
Inkwell	95.60	74.69	83.86
Lomofi	93.40	75.32	83.39
LordKelvin	91.30	81.52	86.13
Nashvile	95.10	78.92	86.45
Rise	4.00	100	7.69
Sierra	4.00	100	7.69
Sutro	4.00	100	7.69
Toaster	4.00	100	7.69
Valencia	93.30	75.24	83.30
Walden	93.50	75.40	83.48
XproII	90.30	83.61	86.83

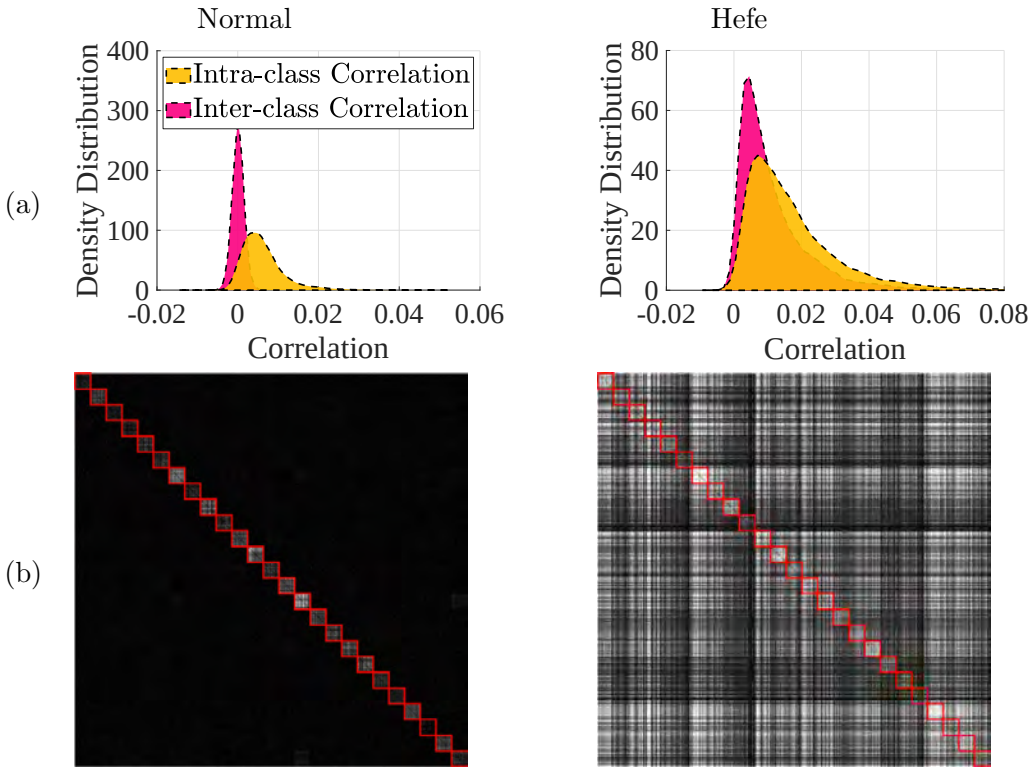


Figure 5.3: Comparison of the pairwise correlations between images with no filters applied ('Normal') and between images filtered by 'Hefe' filter. (a) Distributions plot for the pairwise intra- (yellow) and inter-class (red) correlations from 25 different cameras. (b) Visualization of the pairwise correlations for images from 25 different cameras. The intra-class correlations are delimited by red squares. The brighter color indicate larger correlation values.

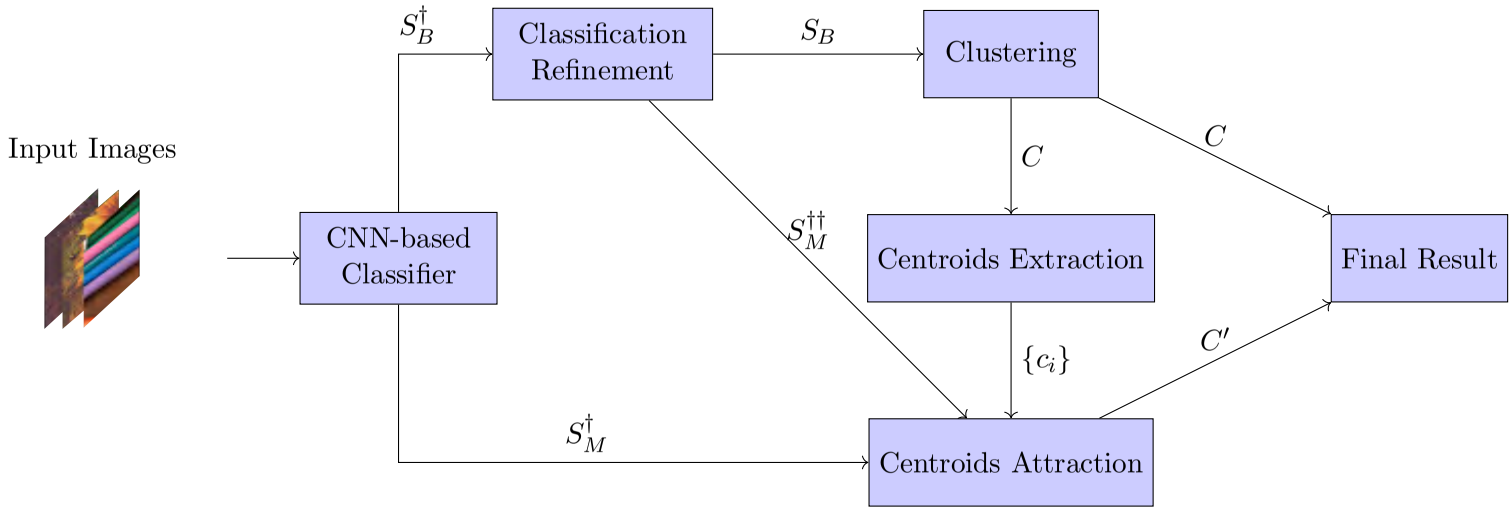


Figure 5.4: Flowchart of the proposed method for PRNU-based source oriented clustering on Instagram images

clusters to process the images in Group M. We, therefore, propose a three-step strategy for the SOC on Instagram images. In the first step, a classifier is constructed for *filter-oriented* image classification to separate the images processed by Group B filters from the rest. In the second step, SOC is performed only on the images classified as processed by the filters in Group B. In the final step, we use the centroids of the clusters discovered in the second step as the reference PRNUs to identify the source cameras for the remaining images, similarly to the task of SCI as described in Section 5.2.1.

The three steps of our proposed framework are illustrated in Fig. 5.4. Specifically, we first pass the images to a convolutional neural network (CNN) based classifier to identify the image filter that has been applied to each image. Based on the classification result, we can separate the images into two sets, $S_B^\dagger$ and $S_M^\dagger$ for the images filtered by a filter from Group B and M, respectively. Due to classification errors, there might be images filtered by a Group M filter left in $S_B^\dagger$. To further purify $S_B^\dagger$, we refine the images $S_B^\dagger$ by comparing the pairwise correlations and the number shared nearest neighbours (SNN) [147] for images in $S_B^\dagger$. If we found that some images in $S_B^\dagger$ are more likely to be from $S_M^\dagger$, we will remove them (i.e. $S_M^{\dagger\dagger}$, a set of images identified during the classification refinement process which may contain both Group M and Group B filtered applied images) from $S_B^\dagger$ to form a purified S_B. After that, we apply the clustering algorithm to the images in S_B to find the set of clusters C. Using the centroids of the clusters in C as the reference PRNUs $\{c_i\}$, we can approach the clustering as a SCI problem by attracting the images remained in $S_M^\dagger$ and $S_M^{\dagger\dagger}$ with $\{c_i\}$ to form the final clustering result. We will present the details about the CNN-based classifier and the classification refinement step in the following parts of this section.

5.3.1 CNN-based Instagram Filter Classifier

The proposed method mainly mitigates the negative impact of the filters in Group M by segregating the images according to the filter classification result. Thus, the performance of the classifier is key to the proposed framework and the classifier needs to be designed carefully. As the Instagram filters may differ from each other greatly, manual feature engineering requires a great amount of study for each filter and the fixed definition of image features might not be helpful when we need to deal with forthcoming filters that are not covered by this study. Moreover, the artifacts introduced by the filters can be content dependent, which may result in very different artifacts for the same filter. Therefore, we use a Convolutional Neural Network (CNN) based classifier to automatically extract features for the filter-oriented image classification task. The CNN architecture used in this chapter takes inspiration from the

well-known Very Deep Neural Networks (VGG-net) [148], which has shown great performance on different image classification tasks. Particularly, Gatys *et al.* [149] manage to use VGG-net to extract and transfer the artistic styles of an image from one artwork to another, which is similar to adding visual effects to an image by applying Instagram filters. This shows that the network architecture is capable of extracting the features from dissimilar styles and inspires us to adopt a similar network architecture for classifying Instagram filters.

The network architecture used in this chapter is shown in Fig. 5.5. It consists of 7 convolutional layers for feature extraction and 3 fully connected layers for classification. Batch normalization is applied to all the hidden layers. The input size of the network is set to $1080 \times 1080 \times 3$, which is the default image size of Instagram. As we aim to classify the images into 18 different classes, the network produces a vector of 18 elements. Softmax function is applied to the vector such that each element in the vector represents the probability of the corresponding image filter being applied to the input image. The network design shares a few similar characteristics with the VGG-net. The VGG-net features small kernel size for the convolutional layers (e.g., 3×3 pixels). This enables the convolutional layers to focus on microscopic features such as texture. Combined with a large number of layers resulting in a large receptive field, the network can extract the macroscopic feature such as color tone at the same time. This makes VGG-net an ideal choice to distinguish the filters. However, the requirement of large input size makes directly adopting the ordinary VGG-net very memory-consuming. Hence, our proposed network has two major differences compared to the ordinary VGG-net. The first difference is that the number of channels for each layer in the proposed network is much smaller than that used in VGG-net. Secondly, in our proposed network, each convolutional layer is followed by a max-pooling layer with a stride of 2. The max-pooling layers help the network extract features more efficiently and the input size of each layer is reduced significantly as the network gets deeper. With these two modifications in place, the memory consumption and the computational cost are significantly reduced, making the training of the network more practicable.

5.3.2 Image Filter Classification Refinement Based on SNN-Correlation Difference

Though a CNN-based classifier is proposed to distinguish Group M and B image filters and its effectiveness can be seen in the following section, its imperfect accuracy does not guarantee a complete separation between images with filtered by Group M and B filters. Thus, some images with Group M filters

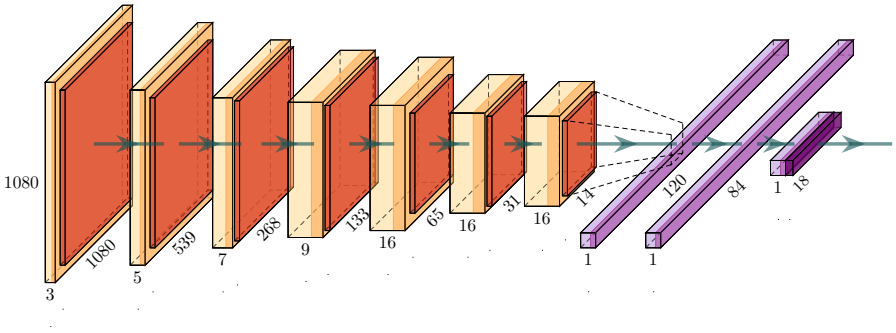


Figure 5.5: The network architecture of the proposed filter-oriented image classifier. The network takes $1080 \times 1080 \times 3$ images as input and outputs a vector of length 18 for the classification. The network consists of 7 convolutional layers (shown in yellow) and 3 fully connected layers (shown in purple). In addition, every convolutional layer is followed by a max-pooling layer. The kernel size for the convolutional layers is 3×3 pixels throughout the network. The number at the bottom is the number of channels for the layer while the number at the sides are the dimension of the layer.

applied could remain in $S_2^\dagger$, which can affect the performance of the proposed clustering method. For example, a cluster contains a significant proportion of images processed by a Group M filter, the centroid the cluster is more likely to mistakenly attract images processed by the same filter later. Thus, to alleviate this problem, a classification refinement step is proposed below.

The main challenge we face by the inclusion of Group M filter applied images in $S_B^\dagger$ is that they may falsely increase the correlations between inter-class images and ultimately bring the risk of grouping the inter-class images into the same cluster. However, for these falsely increased inter-class correlations, the image pairs corresponding to them may share very different neighbours with each other. Figure 5.6 shows three clusters of images from three different cameras (represented by three orange circles) and four images (node a, b, c and d) filtered by the same Group M filter. The dashed lines between a, b, c and d indicate the correlations between them might be falsely increased due to the same applied filter. Statistically, the intra-class correlations should be higher than the inter-class correlations which makes the intra-class image pairs to be closer neighbours to each other. As a result, even though a and b may have a large correlation between them, these two images share very few close neighbours with only c and d as the shared neighbours. It gives us a clue that the disagreement between the pairwise correlations and SNN [147] can be used to discover the images with Group M filters applied left in $S_B^\dagger$. Thus, we aim to find the image pairs with large correlation between their noise residuals but sharing few neighbours by the measure of correlation distances. More specifically, we remove the ith and jth image from $S_B^\dagger$ if $\rho_{ij} > \tau_1$ and

$s_{ij} < \tau_2$, where ρ and s are the pairwise correlation matrix and the SNN matrix, respectively. τ_1 and τ_2 are the two threshold determined from the estimated intra-class correlations and intra-class SNN for each image. To estimate the intra-class correlation and SNN values, we follow the method from [25] using k-means clustering with k set to 2 to differentiate the pairwise correlations and SNNs. Empirically, we set τ_1 to the top 5% of the intra-class correlations and set τ_2 to the smallest value of the intra-class SNNs.

In the demonstration, we assume that the Group M filter applied images left in $S_B^\dagger$ are from multiple cameras and there are only a few of them in $S_B^\dagger$. Apparently, it is not always the case as described by these two assumptions and they may not hold. However, when these two assumptions do not hold, though the proposed method may become less effective, its mechanism of finding the obvious disagreement between the pairwise correlations and SNNs prevents it from repeatedly removing Group B filter applied images and deteriorate the performance of the clustering step. Thus, it is beneficial to apply the proposed refinement to $S_B^\dagger$ to purify $S_B^\dagger$ after the classification step.

5.4 Experiment

5.4.1 CNN-based Instagram Filter-oriented Image Classifier

We first perform a comprehensive evaluation for the proposed CNN-based Instagram filter-oriented image classifier before using it in our proposed three-step SOC framework. As mentioned in Section 5.1, we generate a dataset by filtering $5,370$ images of 25 different source devices from the VISION image dataset using 18 different Instagram filters, which results in a dataset $\mathscr{D}$ consisting of $96,660$ images. These images are divided into training, validation and test sets with a ratio of 60%:20%:20% by randomly selecting an equal number of images filtered by each filter. The proposed network is trained on a desktop with an Intel Core i7-9700K CPU and a Nvidia Geforce RTX 2080 Ti GPU. The special design of the network significantly reduces the consumption of GPU memory, which allows us to train the neural network with a batch size of up to 64. The validation set was used for the tuning of the hyperparameters for the training process. For the rest of this chapter, we will report the results generated with the classifiers trained with a batch size of 64. We train the classifier for 50 epochs using cross-entropy loss and a learning rate of 2×10^{-3} with a SGD optimizer.

Instead of altering the semantic content of an image, most Instagram filters change the image's visual style and introduce different levels of textures, which mainly affects the high-frequency components of the image, where PRNUs reside. This motivates us to investigate the contributions of the image content

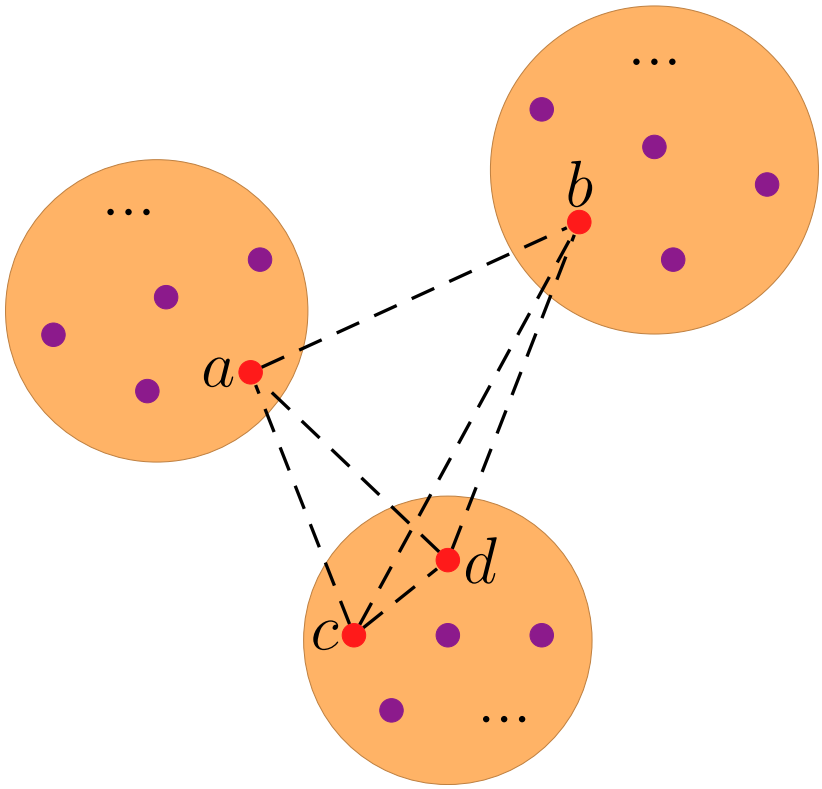


Figure 5.6: A demonstration of how the proposed filter classification refinement method may discover the images filtered by Group M filters remained in $S_B^\dagger$. Each node in the figure represents a candidate image to be clustered and the three circles represents the three ground truth clusters these images belonged to. Nodes a, b, c, d are four images with the same Group M filter applied. Dashed lines are used to indicate the correlations between them may be falsely increased due to the filter.

Table 5.5: The precision ($\mathcal{P}$), recall ($\mathcal{R}$) rates and $F1$-measures for different filters from the proposed CNN-based filter-oriented image classifier trained with different inputs ($\boldsymbol{I}$-net, $\hat{\boldsymbol{I}}$-net and $\boldsymbol{n}$-net). The best precision, recall rates and $F1$-measure for each filter are marked by gray background.

Filters	$\mathcal{P}(\%)$			$\mathcal{R}(\%)$			$F1$-measure(%)		
	$\boldsymbol{I}$-net	$\hat{\boldsymbol{I}}$-net	$\boldsymbol{n}$-net	$\boldsymbol{I}$-net	$\hat{\boldsymbol{I}}$-net	$\boldsymbol{n}$-net	$\boldsymbol{I}$-net	$\hat{\boldsymbol{I}}$-net	$\boldsymbol{n}$-net
Normal	52.89	76.70	96.61	65.55	58.85	98.04	58.54	66.60	97.32
1977	86.47	92.03	91.34	79.14	93.48	95.25	82.64	92.75	93.25
Amaro	83.74	85.54	87.49	69.55	79.33	87.90	75.99	82.32	87.69
Brannan	72.90	80.81	93.44	94.41	88.64	88.92	82.27	84.54	91.12
Earlybird	84.90	88.48	93.10	85.85	95.16	94.23	85.37	91.70	93.66
Hefe	83.00	81.96	92.10	85.94	91.81	94.41	84.44	86.61	93.24
Hudson	87.58	95.97	98.60	91.90	93.11	98.14	89.69	94.52	98.37
Inkwell	94.10	93.22	99.53	56.42	92.18	97.77	70.54	92.70	98.64
Lomofi	58.88	71.12	81.96	69.46	72.91	89.66	63.73	72.00	85.64
LordKelvin	90.19	95.41	98.22	94.97	94.88	97.86	92.52	95.14	98.04
Nashville	81.76	92.58	94.67	85.57	95.25	92.55	83.62	93.90	93.60
Rise	80.32	81.71	87.77	64.62	79.05	85.57	71.62	80.36	86.66
Sierra	72.44	79.93	97.84	77.09	86.41	96.83	74.69	83.04	97.33
Sutro	87.21	91.25	97.91	88.27	96.18	91.62	87.74	93.65	94.66
Toaster	98.21	96.93	97.13	92.27	96.93	97.77	95.15	96.93	97.45
Valencia	66.11	82.74	89.92	69.55	69.65	89.66	67.79	75.63	89.79
Walden	91.47	95.08	96.46	88.83	95.44	96.28	90.13	95.26	96.37
XproII	87.78	88.27	90.77	78.96	91.81	90.69	83.14	90.01	90.73

itself and the high-frequency components (noise residual) to the classification result. Thus, we pre-process the $96,660$ images and generate two more versions of input to the network, namely the denoised image and the noise residual of the image. Again, we use BM3D denoising algorithm [129] to generate the denoised version of the images and extract the noise residuals from three color channels of each image. In such a way, $\boldsymbol{n}$ will have the same dimension as $\boldsymbol{I}$ and $\hat{\boldsymbol{I}}$, which allows them to be fed to the network without changing the network structure. Finally, we train three networks with these three different inputs, namely $\boldsymbol{I}$-net for the original images, $\hat{\boldsymbol{I}}$-net for the denoised images and $\boldsymbol{n}$-net for the noise residuals.

The precision $\mathcal{P}$, recall $\mathcal{R}$ rates and $F1$-measures for 18 filters are reported in Table 5.5. Interestingly, we notice that $\boldsymbol{n}$-net, which takes the noise residuals as the input, outperforms the other two networks for almost all image filters. Though for some filters, $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net have higher precision or recall rates than $\boldsymbol{n}$-net, the performance gap is very small (within about 1% for $\mathcal{P}$ and $1\% \sim 5\%$ for $\mathcal{R}$). Compare the performance based on the $F1$-measures, $\boldsymbol{n}$-net shows consistently better performance apart from the result on 'Nashville' filter. And the performance on Nashville filter from $\boldsymbol{n}$-net is very close to the best performed $\hat{\boldsymbol{I}}$-net. Furthermore, both $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net have problems identifying 'Normal' images, which are the unedited original images. In comparison, the $\boldsymbol{n}$-net has a precision rate of 96.61% and 98.04% for the 'Normal' class. The

Table 5.6: Confusion matrix for the classification of Group M and B applied images produced by the proposed CNN-based filter-oriented image classifiers.

Real/Predict	$\boldsymbol{I}$-net		$\hat{\boldsymbol{I}}$-net		$\boldsymbol{n}$-net	
	M	B	M	B	M	B
M	0.889	0.111	0.925	0.075	0.985	0.015
B	0.047	0.953	0.048	0.952	0.010	0.990

high performance of $\boldsymbol{n}$-net can help the forensic investigators to better identify unedited images. Overall, the $\boldsymbol{n}$-net achieves a precision of 93.52% for all filters while $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net reach 79.92% and 87.29%, respectively. The high precision of $\boldsymbol{n}$-net shows the effectiveness of the proposed CNN-based classifier. We also show the confusion matrix for the classification of Group M and Group B filters as a whole in Table 5.6. Again, $\boldsymbol{n}$-net shows superior performance with only 1.5% of the images in Group M misidentified. Due to the better performance of $\boldsymbol{n}$-net, we will use it as the filter-oriented image classifier for the following experiments of this chapter.

Despite the proposed network's high accuracy on filter classification, we have concerns about the generalization of the network to new *cameras* and *filters*. First, in many realistic forensic scenarios, the training and test images are quite unlikely to be from the same cameras. If a trained network is overfitted to the cameras in the training set, it will not perform well on the images from another set of cameras. To show that our trained network is not overfitted to the cameras in the training set, we test the trained $\boldsymbol{n}$-net on images captured by 11 different cameras of the Dresden Image Database [116]. We form a testing dataset $\mathscr{D}_{\text{Dresden}}$ with 18 different versions for each image from the cameras by applying the 18 different filters, resulting in a total of $29,700$ images. The classification results on $\mathscr{D}_{\text{Dresden}}$ are shown in Table 5.7, where $\boldsymbol{n}$-net shows similar performance as on the images from the VISION dataset, confirming that the trained model is not overfitted to cameras in the training set.

Secondly, new filtering features of Instagram are being developed continually. Thus, despite the 18 filters could be representative for studying the impact of filters on provenance analysis, we would like the classifier to be adaptive and robust to the filters that are not included in the training set. Thus in this experiment, we aim to show that the proposed network trained on a certain number of filters can be easily adapted for other filters by applying transfer learning. We test the $\boldsymbol{n}$-net by training it with images processed by 10 filters first and then apply transfer learning to the trained network to make it available for images processed by other filters as well. To facilitate transfer learning, we change the length of the last layer of the network to match the number of filters the network needs to predict for and keep the rest of the structure

Table 5.7: Filter classification result on images from Dresden Image Database predicted by $\boldsymbol{n}$-net trained with images from VISION dataset.

Filters	$\mathcal{P}$ (%)	$\mathcal{R}$ (%)	F1-measure (%)
Normal	97.28	99.76	98.50
1977	87.34	87.02	87.18
Amaro	90.63	89.75	90.19
Brannan	92.34	83.32	87.60
Earlybird	93.22	92.54	92.88
Hefe	95.65	84.05	89.48
Hudson	99.08	98.18	98.63
Inkwell	99.70	99.82	99.76
Lomofi	66.11	91.42	76.73
LordKelvin	92.98	88.36	90.61
Nashvile	87.70	97.33	92.27
Rise	87.80	90.78	89.27
Sierra	99.11	94.78	96.90
Sutro	93.47	94.60	94.03
Toaster	95.14	98.85	96.81
Valencia	90.18	90.24	90.21
Walden	95.82	94.48	95.15
XproII	98.00	74.41	84.59

unchanged. The weights for the first five convolutional layers are fixed. The weights for the remaining layers are updated by training the network with images, including the ones filtered by the filters not included in the original training set, for another 10 epochs. The performance of the network is shown in Table 5.8. It shows that despite the a disproportional change of number of filters from 10 to 18, the F1-measure remains at a reasonably high level, indicating that the network is able to extract generalised features for the filters by training on only a small number of filters.

Table 5.8: Filter classification results on images from different number of filters by the proposed CNN-based classifier with transfer learning applied. The base model of the classifier is trained with images from 10 different filters.

Number of filters	$\mathcal{P}$(%)	$\mathcal{R}$(%)	F1-measure(%)
10	97.63	97.61	97.62
12	95.42	95.41	95.41
14	94.12	93.99	94.06
16	90.86	90.82	90.84
18	89.69	89.54	89.61

5.4.2 Classification Refinement

In this section, we are going to test the performance of the proposed classification refinement method. We test the proposed method by performing clustering on image datasets of different sizes. First, we construct 5 image datasets with 900, 1350, 1800, 2250 and 2700 images, respectively. For each image dataset, we have equal number of images randomly chosen from 25 source devices and from 18 different filters. Thus, with each filter, each camera accounts for 2, 3, 4, 5 and 6 images for the above mentioned four datasets. We name the five datasets as $\mathscr{D}_2$, $\mathscr{D}_3$, $\mathscr{D}_4$, $\mathscr{D}_5$ and $\mathscr{D}_6$ for convenience.

As we have seen from Section 5.4.1, the proposed CNN-based filter classifier may leave about 1.5% of Group M filter applied images in $S_B^\dagger$. Thus, to ensure the misclassified images that have been processed by Group M filters would not contaminate the cluster centroids extracted after the clustering step and worsen the performance of the ensuing centroid attraction, the performance of the proposed filter classification refinement step can be critical. Figure 5.7 illustrates the performance of the proposed filter classifier and the classification refinement method over the test datasets. First, we notice as we have seen from Section 5.4.1, the classifier's performance is satisfactory even for the biggest dataset, $\mathscr{D}_6$, with 2700 images in total and 1050 Group M filter applied images. Only 18 Group M filter applied images are misidentified and included in $S_B^\dagger$.

To apply the proposed classification refinement method, the pairwise correlation matrices and the SNN matrices for the datasets were computed. To compute the pairwise correlations, we use the green channel of the full-sized noise residuals from each image. For the computation of the SNN matrices, we compare the 20 nearest neighbours of each image between the image pairs. Following the method proposed in Section 5.3.2, the number of Group M filter applied images removed from $S_B^\dagger$ is plotted in yellow as shown in Fig. 5.7. The total number images in $S_M^{\dagger\dagger}$, which is the set of the images removed from $S_B\dagger$ by the refinement method, is plotted in red for each tested dataset. From Fig. 5.7, as it has been discussed in 5.3.2, we can see clues indicating that the performance of the proposed refinement method is less effective when the number of Group M filter applied images are too small (e.g. $\mathscr{D}_2$) and Group M filter applied images become less sparse in $S_B^\dagger$ (e.g. $\mathscr{D}_6$). Overall, as the yellow line shows that a large portion of Group M filter applied images are correctly identified from the one left in $S_B^\dagger$ (shown by the blue line), it shows the proposed refinement method is effective in reducing the number of Group M filter applied images in $S_B^\dagger$. As a result, the subsequent clustering and centroids attraction steps from the proposed three-stage clustering framework can be less affected by the Group M filters.

Another aspect worth mentioning is that though the proposed classification

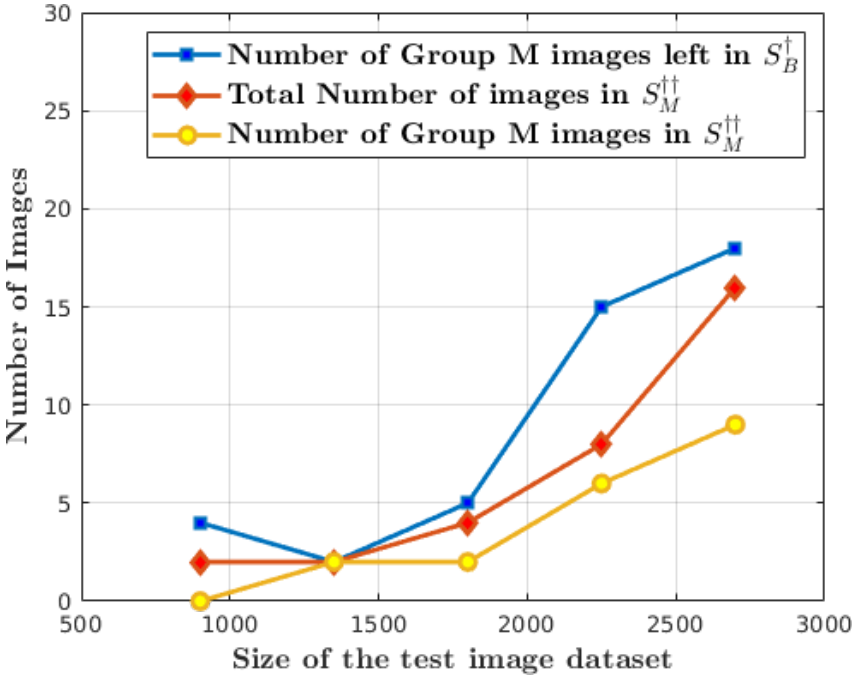


Figure 5.7: The performance of the proposed CNN-based filter classifier and the classification refinement method tested on image datasets of different sizes.

refinement step may also remove some Group B filter applied images from $S_B^{\dagger}$ as some Group B filter applied images may show strong disagreement between the pairwise correlations and SNNs (e.g. a correlation can be unexpectedly high due to the randomness of the correlation distribution), it is not a serious problem. First, the number of images being removed is small comparing to the total number of Group B filter applied images to be clustered (e.g. 8 images falsely removed from 1650 Group B filter applied images in $\mathscr{D}_6$). More importantly, by applying the proposed refinement step, the centroids extracted from the clusters can be less contaminated by the Group M filters, which makes them more representative for the source device each cluster accounts for. With each centroid better representing the source devices in the test dataset, the wrongly removed Group B filter applied images can have greater chance being attracted to the right cluster during the centroids attraction step. Overall, by testing over different datasets, the effectiveness of the proposed classification refinement step is proved.

5.4.3 Source-oriented Clustering of Instagram Images

After testing the effectiveness of the proposed CNN-based image filter classifier and the classification refinement method, we test the overall performance of the proposed three-stage clustering framework with the five datasets mentioned above. We use the SOC method in [25] to perform the clustering step as described in Section 5.3. The centroids for each cluster are calculated by

Table 5.9: The performance of the proposed three-step clustering framework on 5 Instagram image dataset of different sizes. The figures in the table are presented in percentage.

Dataset	No. of Images	$\mathcal{P}$ (%)	$\mathcal{R}$ (%)	F1-measure (%)
$\mathscr{D}_2$	900	87.39	87.41	87.40
$\mathscr{D}_3$	1350	93.73	83.75	88.46
$\mathscr{D}_4$	1800	95.72	85.52	90.33
$\mathscr{D}_5$	2250	94.57	76.28	84.45
$\mathscr{D}_6$	2700	95.49	77.02	85.26

averaging the noise residuals of the images in the cluster. Table 5.9 shows the precision, recall and F1 measure for the proposed framework on the five test sets, the same ones as in Section 5.4.2. Though the performance varies slightly across different datasets, the framework is able to obtain F1 measures over 80% for all of the five test sets. The consistently high F1-measures show the effectiveness of the proposed framework. Comparing the performance of the proposed framework over $\mathscr{D}_4$ in Table 5.9 with the results from Section 5.2.2, which was obtained on the same set of images, by applying the same clustering method proposed by [25] without using the three-step clustering framework, an overall improvement in both precision and recall rate can be observed. Thus, despite the Group M filters may contaminate the PRNUs embedded in the images, the proposed three-step clustering framework provides a practical solution to perform PRNU-based SOC on Instagram images.

5.5 Conclusion

With built-in image editing tools like 'filters' on Instagram becoming a common practice on SNSs, these tools ultimately pose new challenges to PRNU-based forensic investigations. In this chapter, using Instagram filter as an example, we took a close look at the impact of these image editing tools on PRNU-based source camera identification (SCI) and source-oriented clustering (SOC). We discovered that though PRNU-based SCI remains effective for filtered images on Instagram when quality reference PRNUs are available, the artifacts introduced by certain Instagram filters can severely affect the performance of PRNU-based SOC as there is no reference PRNU. To address this problem, we proposed a three-step clustering framework. As a main component of the framework, a CNN-based filter-oriented image classifier is proposed and it achieves an overall 93.52% precision in identifying the filters applied to images. We have also shown that the proposed CNN architecture generalises well on new cameras and image filters. With the success of the filter-oriented image classifier, the proposed three-step clustering framework achieves an F1-measure of 90.33% in

SOC, which is a significant improvement compared to the F1-measure 47.74% obtained by directly applying existing clustering methods on Instagram images. Thus, the framework provides a practical solution for the provenance analysis of user-edited images on SNSs.

Chapter 6

Detecting Anti-forensics Attacks on PRNU Using Generative Adversarial Networks

As PRNU has been widely used for different tasks in digital image forensics, it also becomes a target for anti-forensics attacks. As we have discussed in Chapter 2, one effective way to carry out these anti-forensics attacks is through the suppression of PRNU using image manipulations like denoising or median filtering. PRNU-based provenance analysis is no longer feasible when the image in question has undergone such attacks. Thus, detecting whether an image's PRNU is attacked or not is an important topic in digital image forensics. Section 2.4.2 shows that the convolutional neural network-based method can extract residual-based features to detect different manipulations used for anti-forensics attacks on PRNU. However, we found that though these CNN-based detectors can perform well in terms of detecting certain types of manipulations, the networks' excessive emphasis on manipulation-specific features could be a problem. For example, this may prevent them from generalising well for the binary classification task of detecting whether an image's PRNU is attacked or not, especially when they encounter images attacked by manipulations not included in their training set. To address this issue, we propose a generative adversarial networks-based training framework to help the trained network generalise better for the binary classification task. Experimental results show that the proposed GAN-based training framework can help the classifier performs better on detecting unprecedented attacks.

The reminder of this chapter is organised as follows. In Section 6.1, we will briefly introduce the background of anti-forensics attacks on PRNUs and

generative adversarial networks. In Section 6.2, the details of the proposed generative adversarial networks-based training framework will be given. Section 6.3 presents the experiments done to validate the effectiveness of the proposed framework. Finally, Section 6.4 concludes the chapter.

6.1 Background

In the previous chapters, we have shown how the PRNU can be used as an effective tool for different multimedia forensics tasks like source camera identification, source-oriented image clustering, and image forgery detection. Despite the advantage and popularity of using PRNU for multimedia forensics, being a noise-like signal makes it vulnerable to different anti-forensics attacks. Residing in the high-frequency domain, the PRNU can be easily attenuated or removed even by simple manipulations, including Gaussian blurring and medium filtering, etc. These operations can either be carried out deliberately by attackers to hide provenance of the image or just by unintended over-compression. It is acknowledged in [20] that aggressive denoising filters could suppress PRNUs and prevent PRNU-based source camera identification. Sengupta *et al.* [108] use a median filter to anonymise the images. In [109], Villalba *et al.* suppress the PRNUs by using a combination of the wavelet transform and Wiener filter. In general, these methods use noise filtering to attenuate PRNUs. We also tried other denoising filters like Gaussian filter and BM3D [129] and found they could effectively suppress PRNUs as well by applying them aggressively.

When we perform PRNU-based forensic methods on the attacked images, not only will extra computational costs be required, forensic investigators may also be misled to make wrong conclusions. Using source-oriented clustering as an example, this task usually requires the computation of the pairwise correlations between the PRNUs extract from the images. The computation complexity is $\mathcal{O}(n^2)$ with respect to the number of images. Thus, including any images with PRNU absent or attacked would significantly increase the computational cost while not providing any useful information. Furthermore, these flawed images should be viewed as outliers in any cluster. With some existing algorithms being particularly sensitive to outliers [25, 98], these images could notably downgrade the clustering performance. Therefore, being able to identify the images subject to anti-forensics attacks before applying the PRNU-based methods is important.

Neural network-based methods have been widely applied in various fields of computer science. Especially with the emergence of deep neural network structures like VGG [148] and residual network [150], the superior ability to extract non-trivial features compared to hand-crafted features boosts the popularity of the neural networks. As mentioned in Section 2.4.2, inspired by

the use of residual image extracted through high-pass filtering in [112, 115], Bayar and Stamm [77] proposed a manipulation detection method using a constrained convolutional neural network (CNN) architecture with the first convolutional layer forced to perform high-pass filtering. Cozzolino *et al.* further investigate the relationship between residual-based descriptors and CNN in [78]. They found that there is no real contraposition between the residual-based features and CNNs, even for unconstrained CNN architectures. Besides, it is demonstrated that their proposed CNN architecture manages to detect various manipulations accurately, including Gaussian blurring, median filtering and JPEG compression. A similar CNN architecture capable of detecting multiple manipulations is presented in [79].

With the powerful performance from the neural network-based classifiers, we can treat the task of detecting anti-forensics attacks as an image classification problem with neural networks. A neural network-based classifier can be trained by feeding both pristine and attacked images to the network. However, issues may arise due to the limitation of the training set. Given a training set of finite size, it may only cover images subject to certain types of attacks. A neural network model, trained with a dataset like this, may perform well in terms of extracting features related to these manipulations. However, when the network is applied to images attacked by unprecedented operations, the network's performance cannot be guaranteed. This is because the network put too much focus on the predominant features corresponding to certain types of manipulations presented in the training set. As a result, the network could miss the semantics of the ultimate goal: detecting whether an image's PRNU is pristine or not.

To address this problem, in this chapter, we propose a novel training strategy by using generative adversarial networks (GAN) for training data augmentation. GAN is first proposed by Goodfellow *et al.* in [151]. Composed of a pair of networks, namely a generator and a discriminator, the generator can capture the statistical properties of the real samples and improve the generated samples through the adversarial process against the discriminator. Correspondingly, the discriminator will learn the difference between the real and generated samples, forcing the generator to improve. Kim *et al.* use GAN in [152] to hide the trace of median filtering by learning the statistical characteristics of the pristine images.

Different from the existing GAN-based methods for anti-forensics attacks by generating images with reference to the pristine images, our proposed method presented in this chapter trains the generator without direct knowledge on the pristine images. The 'pristineness' of the generated images is entirely learnt from the adversarial process. In addition, instead of aiming for a generator that can fool the discriminator completely, we moderate the generator's training

process by setting an intermediate goal for the adversarial process, forcing the generated images into a mixed state of the pristine and attacked images from the original training set: the generated images can be considered as having their PRNUs attacked as the attacked images from the original training set; meanwhile, the same as the pristine images, the generated images do not possess predominant features corresponding to the types of manipulations included in the original training set. In this way, we are not trying to generate pristine images from the attacked images. Instead, we generate 'lightly attacked' image to help the classifier build a better understanding of the attacked images. The generated images can divert the excessive attention, paid by the discriminator on manipulation-specific features included in the original training set, to the semantics of detecting whether an image's PRNU is attacked or not. The proposed method manages to improve the trained classifier's performance on unprecedented attacks.

6.2 Proposed Method

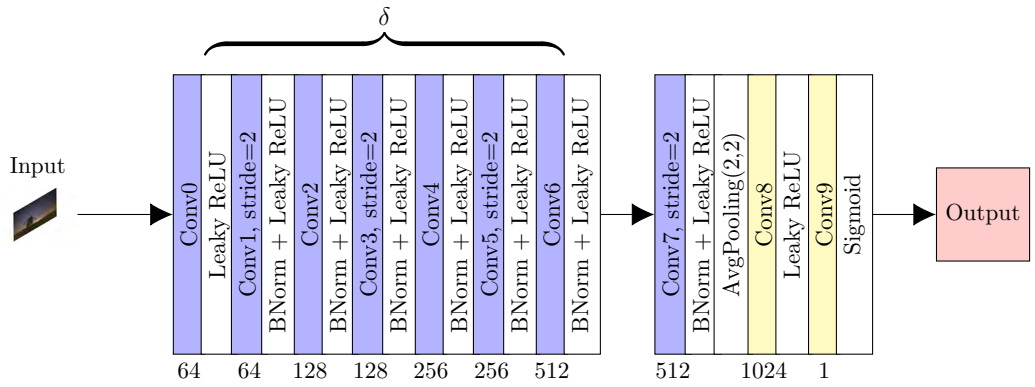


Figure 6.1: The network structure of the proposed classifier, which is working as the discriminator, $\mathcal{D}$, in the proposed GAN framework. All the convolutional layers shown in blue have kernel size of 3×3. The convolutional layers shown in yellow has kernel size of 1×1. The number below each convolutional layer represents the number of the output channels from the layers. The layers included in the bracket are the feature extraction layers of the network, marked as δ. The output of the network is a real number in the range of $[0, 1]$.

The goal of this work is to design a training framework to build a classifier which could identify the images with their PRNUs attacked. The classifier should work even when the training set only contains images subject to a specific type of anti-forensics attack. To achieve this goal, we train the classifier with a GAN framework, which consists of two networks, namely a discriminator, $\mathcal{D}$, and a generator $\mathcal{G}$. We consider the scenario when we have two sets of image patches, one set, $\mathcal{P}$, with pristine images and the other, $\mathcal{F}$, with images

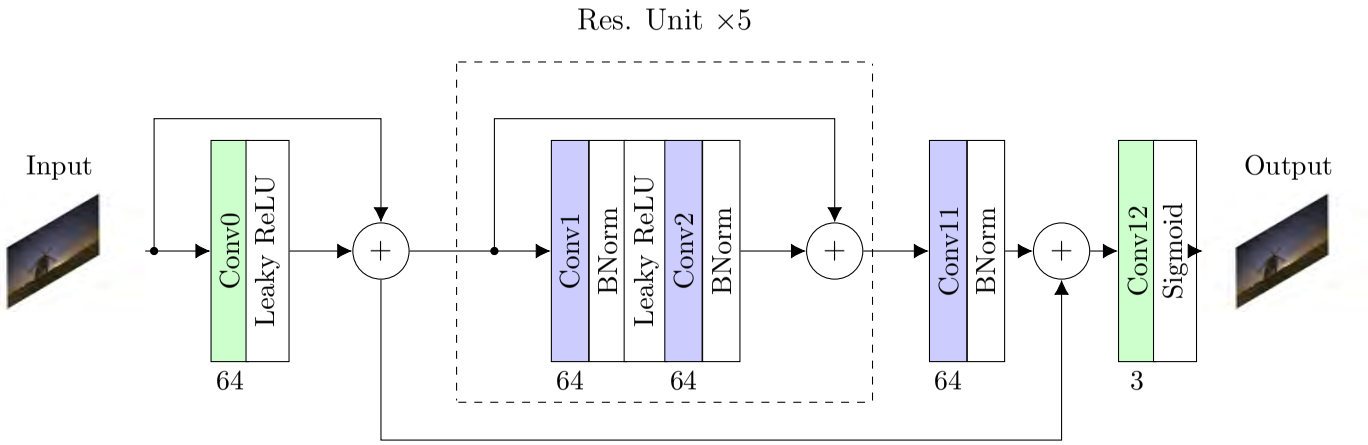


Figure 6.2: The structure of the proposed generator, $\mathcal{G}$, in the GAN framework. The generator follows the main concept of ResNet with multiple residual units. The repetitive units are highlighted in the dashed rectangle. The convolutional layers shown in green color have a kernel size of 9×9 while the layers shown in blue have a kernel size of 3×3. The network takes an image as input and output an image of the same size.

undergone certain types of anti-forensics manipulations. $\mathcal{P}$ and $\mathcal{F}$ have the same number of image patches. PRNUs reside in the high-frequency bands of the images. Thus, regardless of the type of manipulations, any effective attacks on PRNUs need to change the high-frequency components of the images. It is reasonable to assume that the traces of attacks could be found by studying the high-frequency residuals of the images. As mentioned in [78], CNN can perform in the same manner as the residual-based descriptors. With this in mind, we build a classifier following a CNN structure.

The structure of the proposed binary classifier, which also works as the discriminator, $\mathcal{D}$, in the GAN framework, is shown in Fig. 6.1. The structure is mostly convolutional without any Max pooling or fully-connected layers. We use Leaky ReLUs for the activation layers to prevent vanishing gradients. The above designs are in place to improve the stability of the network during the GAN training process. With the Sigmoid function at the end, the network takes an image patch as input and outputs a single number in the range of $[0, 1]$. The output can be considered as a measurement of the similarity between the input image patch and pristine images. We first train the classifier $\mathcal{D}$ with the pristine and attacked images from $\mathcal{P}$ and $\mathcal{F}$ only. Given an input image patch p_i, we label it with l_i:

$$l_i = \begin{cases} 1, \text{if } p_i \in \mathcal{P}, \\ 0, \text{Otherwise} \end{cases} \tag{6.1}$$

We define the loss for the binary classification, $\mathcal{L}_{\text{binary}}$, for each output $\mathcal{D}(p_i)$ as:

$$Ł_{\text{binary}}(\mathcal{D}(p_i)) = |\mathcal{D}(p_i) - l_i| \tag{6.2}$$

A binary classifier can be trained by minimising the loss over inputs from $\mathcal{P}$ and $\mathcal{F}$ using the stochastic gradient descent (SGD) algorithm. We call the classifier trained with this binary data as $\mathcal{D}^*$. As we mentioned earlier, this binary classifier $\mathcal{D}^*$ will focus on the predominant features related to the specific manipulations presented in the training set. As a result, when it is applied to images attacked by unprecedented manipulations which do not possess those features, it could miss the difference between pristine and attacked images. Thus, we introduce the GAN framework to address this problem. We will tune the discriminator $\mathcal{D}$ using images generated from a generator $\mathcal{G}$. Unlike the work in [152] generating images with reference to the pristine images to learn their statistical properties, we want the generated images to possess some statistical properties of the images in $\mathcal{F}$, which can help the classifier better understand the attacked images. Our generator G generates images with reference to the manipulated images from $\mathcal{F}$. Using a manipulated image patch p_i from $\mathcal{F}$ as input and without the information about the patch's PRNU throughout the GAN training process, we can ensure

that the generated image patch $\mathcal{G}(p_i)$ will not possess PRNU and thus, can be considered as attacked. As a 'lightly attacked' image will possess some visual similarity with its attacked version but without the predominant features corresponding to the manipulations the attacked version is subject to, which have been extracted by the feature extraction layers $\mathcal{D}^*$, for the generator $\mathcal{G}$, we define its loss as follows with a goal to minimise it:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}}(\mathcal{G}(p_i)) =& \mathcal{L}_{\text{Visual}}(\mathcal{G}(p_i), p_i) - \mathcal{L}_{\text{Feature}}(\mathcal{G}(p_i), p_i) \\ &+ \mathcal{L}_{\text{Adversarial}}(\mathcal{D}(\mathcal{G}(p_i)))\end{aligned} \tag{6.3}$$

$\mathcal{L}_{\text{Visual}}(\mathcal{G}(p_i), p_i)$ measures the visual difference between the generated patch $\mathcal{G}(p_i)$ and the manipulated patch p_i, which is defined as:

$$\begin{aligned}\mathcal{L}_{\text{Visual}} = &\alpha \cdot |L_2(\mathcal{G}(p_i), p_i) - \Lambda| \\ &+ \beta \cdot L_2(\text{vgg}(\mathcal{G}(p_i)), \text{vgg}(p_i))\end{aligned} \tag{6.4}$$

where $L_2(\cdot)$ measures the L_2 distance between two tensors and $\text{vgg}(\cdot)$ is the feature extraction layers from a VGG16-net pre-trained on ImageNet [153]. α and β are two weight coefficients and Λ is a relaxation term. The term regarding the L_2 between the generated image and the input image in Equation (6.4) mainly measures the pixel-wise difference between the input and the generated image patch. This can put a constraint on the generated image and help the generated image possess similar statistical properties of the manipulated image. However, as we allow the generated image to be slightly different from the input image, we introduce a relaxation Λ for the L_2 distance between the input and generated image patches. The latter term in Equation (6.4) measures the perceptual difference between the two image patches. The pre-trained VGG16-net is capable of classifying different images according to their class labels appeared in ImageNet. Thus, the feature map extracted from the selected layers can be viewed as an abstract measurement of the visual content of an image. By measuring the difference between the patches in the feature space of a VGG16-net, minimizing this term ensures the generated patches share the perceptual similarity with the input patches.

The term $\mathcal{L}_{\text{Feature}}(\mathcal{G}(p_i), p_i)$ measures the difference between the two patches in the feature space of $\mathcal{D}^*$:

$$\mathcal{L}_{\text{Feature}}(\mathcal{G}(p_i), p_i) = \gamma \cdot L_2(\delta^*(\mathcal{G}(p_i)), \delta^*(p_i)) \tag{6.5}$$

where γ is another weight coefficient and $\delta^*(\cdot)$ stands for the feature extraction layers (the part in bracket in Fig. 6.1) of $\mathcal{D}^*$. As the predominant features extracted by $\mathcal{D}^*$ are more likely to be manipulation specific, which might not perform well on the binary classification of deciding whether an image's PRNU

is attacked or not, we want to feed 'attacked' images without these predominant features to the classifier in order to tune it. The negative sign before this term in Equation (6.3) encourages the generated image to be different from the input image in terms of these predominant features.

The adversarial loss is defined as:

$$\mathcal{L}_{\text{Adversarial}}(\mathcal{D}(\mathcal{G}(p_i))) = \eta \cdot |U - \mathcal{D}(\mathcal{G}(p_i))| \tag{6.6}$$

Again, η is a weight coefficient. With the definition of labels given in Equation (6.1), a traditional GAN framework will immediately set U to 1. In this way, the generator $\mathcal{G}$ will optimise its weight to maximise the chance of having the generated images to be labelled as pristine by the discriminator. However, our goal is not to train a generator which can fool the discriminator completely. We want the discriminator $\mathcal{D}$ to be able to eventually figure out the underlying difference between the pristine and attacked images. On the other hand, neither it means we should set U to 0, otherwise there will be no adversarial process. As a result, the generator would mainly just suppress the most dominant features due to the $\mathcal{L}_{\text{Feature}}$ term. As $\mathcal{D}^*$ is a powerful feature extractor, it could extract multiple manipulation-specific features. Even with some of the most dominant features suppressed, other manipulation-specific features might remain strong and the discriminator will only focus on them if there is no adversarial process. Thus, we want to keep a moderate adversarial process: *the generator should be powerful enough to drive the discriminator to explore more features, which can better differentiate the pristine and attacked images; but the generator should not be too powerful such that the generated images will lose all these features.* So U is set to a number between 0 and 1 to moderate the adversarial process.

With the loss function for the generator defined, we follow the same definition for labelling and the loss for the discriminator as the ones defined for the original classifier using Equation (6.1) and (6.2). We fix the weights for the feature extraction layers in $\mathcal{D}$ as δ^* and train the rest of the discriminator with images from $\mathcal{P}$, $\mathcal{F}$ and generated images with a number ratio of $50\% : 25\% : 25\%$. With this ratio, it allows equal number of pristine and attacked images to be fed to the discriminator for training as the generated images should still be considered as 'attacked' images. This ensures that no bias towards one class label is introduced due to the imbalanced number of training images for different labels. We keep using images from $\mathcal{F}$ for training to allow the updated discriminator maintaining its performance in differentiating images from $\mathcal{P}$ and $\mathcal{F}$. We fix the feature extraction layers' weights assuming that the feature extraction performance of $\mathcal{D}^*$ is powerful enough that it has already discovered the features which can differentiate the pristine and attacked images. This is

a reasonable assumption considering the deep structure of the discriminator and the features differentiating pristine and attacked images are helpful for the binary classification between images from $\mathcal{P}$ and $\mathcal{F}$. Thus, $\mathcal{D}^*$ has the ability and the incentive to discover features. By locking the feature extraction layers' weights, it prevents the deep structure of the discriminator from being too powerful and cuts the adversarial process and thus, helps to maintain a more stable GAN training process. With the adversarial process, the discriminator tends to decrease its emphasis on manipulation-specific features and put more weights on the features which can better differentiate the pristine and attacked images. We will use experiment to show the effectiveness of the proposed framework.

6.3 Experiments

To test the effectiveness of the proposed framework, we run experiments on images from the Warwick Image Forensics Dataset [3]. We use images from 8 different cameras and each camera accounts for 200 images. We partitioned the images into $268,180$ non-overlapping patches of size 256×256 pixels and divide them into training and testing sets with a ratio of $90\% : 10\%$ randomly. On the training images, we keep the original images as $\mathcal{P}$ and generate $\mathcal{F}$ by applying strong Gaussian blurring using a Gaussian kernel with a standard deviation of 8 on all images. Such a strong blurring is effective in removing the PRNUs in the images. However, this also leaves obvious visual differences between the attacked and the pristine images. Thus, the neural network-based classifier could easily learn the features associated with this visual effect and differentiate these two types of images. We train the classifier without the proposed GAN framework on the training images for 5 epochs and obtained the weights for $\mathcal{D}^*$. We use the SGD algorithm as the optimizer with a learning rate of 0.04. The training images are loaded in batches with a batch size of 16. Unsurprisingly, this network achieves a high accuracy of 98.10% in differentiating the strong Gaussian blurred and pristine images. However, when we apply this network on images attacked by other manipulations or even just using a Gaussian kernel but with a smaller standard deviation, the classification performance becomes much worse as shown by Figure 6.3(a).

To test the performance of the original classifier on images subject to unprecedented attacks without using the proposed GAN training framework, we applied three types of manipulations on the pristine images to form 3 sets, one with a weaker Gaussian blurring compared to the manipulation applied to the images in $\mathcal{F}$, using a kernel with a standard deviation of 3, one with Median Filter which has a kernel size of 7×7 and the last one with BM3D denoising. For simplicity, we refer to the weaker Gaussian blurring as 'Gaussian Blurring'

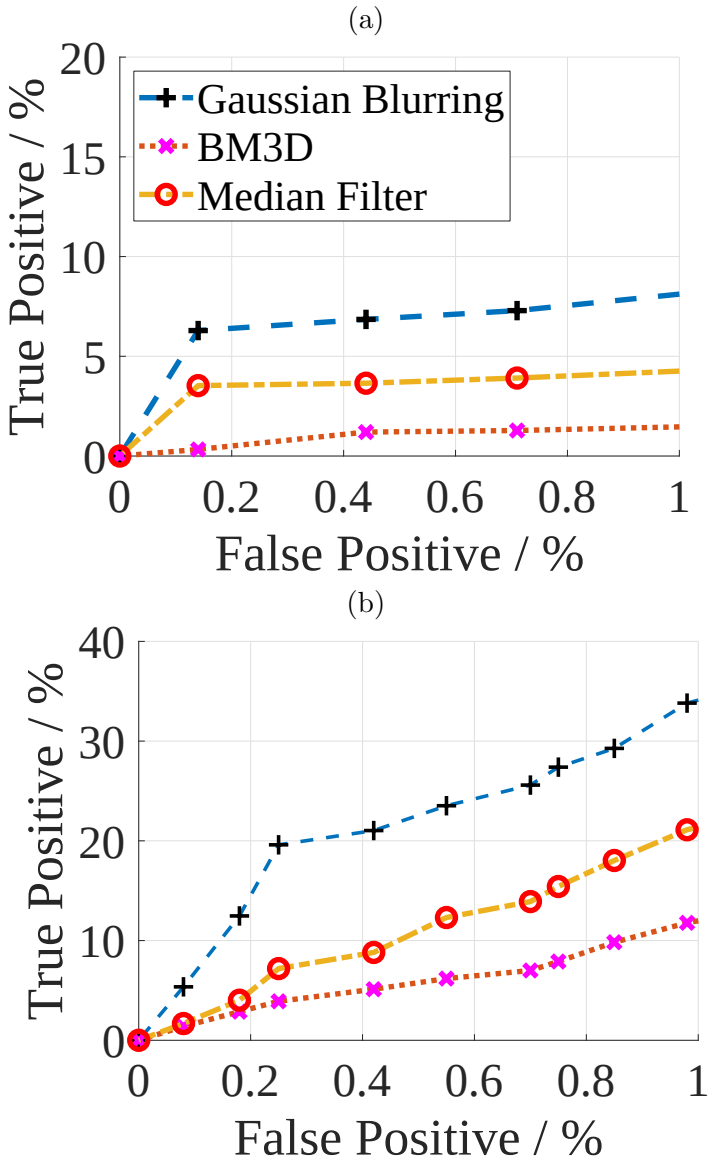


Figure 6.3: ROC curves for the classification results on images attacked by three different manipulations detected by (a) $\mathcal{D}^*$, (b) classifier trained under the proposed GAN framework.

in Fig. 6.3 and use the term 'strong Gaussian blurring' specifically for the manipulation applied on images in $\mathcal{F}$. We use $\mathcal{D}^*$, which is not trained under the GAN framework to classify the images. As $\mathcal{D}^*$ outputs a single number which can be viewed as a measure of similarity, we can adjust the threshold for this similarity to generate detection results with different levels of true positive rates (TP, attacked images labelled as attacked) under different false positive rates (FP, pristine images labelled as attacked). As we want to use this classifier to filter attacked images from pristine images when we apply PRNU-based methods on a large number of candidate images, we would like to have a small FP rate such that we can keep as many pristine images as possible. Thus, for this test, we only focus on the results when FP rate is below 1%. The Receiver Operating Characteristic (ROC) curves for $\mathcal{D}^*$ on the three test sets are shown in Fig. 6.3(a). The classifier has TP rate lower than 5% for images manipulated by median filter and BM3D. While the set with the weak Gaussian blurring is performing better than the other two as this manipulation shares more similarities with the images in the training set, the TP rate is still below 10% throughout the range. It proves the neural network-based classifier can be overfitting to the features specifically related to the strong Gaussian blurring.

To show the effectiveness of the proposed framework, we apply it on $\mathcal{P}$ and $\mathcal{F}$ to tune $\mathcal{D}$. The weight parameters α, β, γ and η are set to 1, 3×10^{-3}, 0.01 and 0.01, respectively. The relaxation term Λ is set to 10^{-3} and the adversarial goal U is set to 0.7. SGD optimizer is used for both the discriminator and the generator with the learning rate set to 0.02 for the generator and 0.01 for the discriminator. We tested different combinations of these parameters using the discriminator's performance on the task of classifying images subject to weak Gaussian blurring and pristine images as a reference. The parameters shown above produce good stability for the training process of the GAN framework and generates the best performance for the above-mentioned classification task among the set of combinations of parameters we tested. Thus, the experimental results reported from this section are all based on the network trained with the above-mentioned parameters. The discriminator and generator are trained in turns. The rule for alternating the network being trained is that after training the discriminator with a batch of 16 images, the generator is trained for two batches of input images with the same batch size. The reason for the generator to be trained for more rounds is that the training of the generator should be considered as more challenging than the training of discriminator for this task. Only by training the generator for more rounds, a stable training process can be obtained. Otherwise, the discriminator would be too powerful for the under-trained generator and when that occurs, the generator might not be improved for the subsequent training process. We trained the networks

under this GAN framework with the discriminator trained for 3 epochs. Each epoch takes around 14 hours to run on a Nvidia RTX 2080Ti GPU. The ROC curves for the tuned classifier, which is the discriminator in the proposed GAN framework, on the detection of the 3 manipulations are shown in Fig. 6.3(b). The proposed framework can achieve TP rate as high as 37.9% with FP rate lower than 1% for the weak Gaussian blurring despite all the attacked images in the training set are manipulated by a much strong Gaussian blurring. Even for the BM3D denoised images, which do not show strong local blurring artifacts, the tuned classifier can detect them better than the original classifier can do for the weaker Gaussian blurring. Compared to the performance of the original classifier, this shows the effectiveness of the proposed framework of shifting the classifier's excessive emphasis on manipulation-specific features to the more generalised features shared by images with their PRNUs attacked. Despite the overall TP rate is not very high, the classifier is still of good forensic significance. Thanks to the low FP rate, it conveniently allows us to identify attacked images by combining the patch-level detection results to image-level. If we found an image with a large number of patches identified as attacked, then we can deem the image as an attacked image with high confidence due to the low FP rate of the classifier.

6.4 Conclusion

Existing neural network-based anti-forensics attack detectors have shown superior ability in detecting specific manipulations on images. However, when they are used for the binary classification of deciding whether an image's PRNU is attacked or not, their performance could be compromised due to their excessive emphasis on manipulation-specific features. In this work, we proposed a novel strategy to tune such a detector, which is in the form of a binary classifier. By training it as the discriminator in a GAN framework, this makes the classifier generalize better for images subject to unprecedented attacks. Despite the limitation of the original training set which might only contains images subject to a certain type of attacks, the generated images in the GAN framework can shift the classifier's excessive focus on those manipulation-specific features to the ones which can generalise better for the binary classification. The experimental results show that the proposed training scheme could improve the probability of detecting attacks on PRNUs from manipulations not included in the training set while keeping the false positive low for the pristine images.

Chapter 7

Conclusions and Future Work

The work presented in this thesis has been concerned with digital image forensics techniques based on photo response non-uniformity (PRNU), which is a powerful device fingerprint. Methods based on PRNU have been successfully applied for source camera identification, source oriented image clustering and image forgery detection. However, these methods also face challenges from different aspects and this thesis aims to address some of these challenges. In this thesis, a novel image forensics dataset, namely the Warwick Image Forensics Dataset is constructed and presented in Chapter 3, with special attention paid to the camera exposure settings. The specially designed dataset enables works to be carried out to investigate the ISO speed's impact on PRNU-based image forgery detection as well as developing method to account for ISO speed's impact. These studies are shown in Chapter 4. The impact from the image filters used by Instagram on PRNU-based source oriented image clustering is studied in Chapter 5 and a three-step clustering framework is proposed. Finally, to help the neural network-based anti-forensics attack detector generalise better for unprecedented attacks on PRNUs, a generative adversarial networks-based training strategy is proposed in Chapter 6 to overcome the limitation of the training data. The following sections summarise the key contributions from the previous chapters. The directions for future research will be discussed in the last section of this chapter.

7.1 Warwick Image Forensics Dataset

Device fingerprints like sensor pattern noise (SPN) are widely used for provenance analysis and image authentication. Over the past few years, the rapid advancement in digital photography has greatly reshaped the pipeline of image capturing process on consumer-level mobile devices. The flexibility of camera exposure parameter settings and the emergence of multi-frame photography algorithms, especially high dynamic range (HDR) imaging, bring new challenges

to device fingerprinting. The subsequent study on these topics requires a new purposefully built image dataset.

With the requirement for carrying out studies on the impact from different camera exposure parameter settings on device fingerprinting, as well as the goal to build a dataset which could facilitate the development of PRNU-based forensic methods for HDR images in mind, the Warwick Image Forensics Dataset is constructed. The dataset consists of more than $58,600$ images, captured with 14 different digital cameras. Compared to the existing datasets, e.g., the Dresden Image Dataset [116] and the VISION dataset [117], the images in Warwick Image Forensics Dataset feature more diverse camera exposure parameter settings. Thus, systematic studies on these parameters could be carried out. In addition to that, the inclusion of images taken using auto exposure bracketing (AEB) and high speed burst functions allow different multi-frame computational photography algorithms, including HDR imaging, to be applied on the images from this dataset. Thus, the special design of this dataset could help studies on various topics in digital image forensics to be carried out in the future.

7.2 Addressing the Impact of ISO Speed Upon PRNU and Forgery Detection

PRNU-based forgery detection methods often reveal manipulated areas by finding the regions with PRNU absent. To check the existence of the PRNU, the correlation between an image's noise residual with the device's reference PRNU is often compared with a decision threshold. A PRNU correlation predictor is a key component to determine this decision threshold by assuming the correlation is content-dependent. However, we found that not only the correlation is content-dependent, but it also depends on the camera sensitivity setting, which is more commonly known by the name of ISO speed.

In Chapter 4, based on the Poissonian-Gaussian noise model from [88], we show how the PRNU correlation is dependent on the ISO speed both analytically and experimentally. Due to this dependency, we postulate that a correlation predictor is ISO speed-specific, i.e. *reliable correlation predictions can only be made when a correlation predictor is trained with images of similar ISO speeds to the image in question.* Thus, we proposed an ISO speed-specific correlation prediction process for PRNU-based image forgery detection. By testing the forgery detection method from [24] following the proposed ISO speed-specific correlation prediction process, more consistent and reliable forgery detection performance is observed for both realistic and synthetic image forgeries compared to the alternative correlation prediction training

process.

Recognizing that in the real-world, information about the ISO speed may not be available in the metadata to facilitate the implementation of the ISO speed-specific correlation prediction process, we propose a block-matching based method to infer an image's ISO speed from the image content, called CINFISOS (Content-based Inference of ISO Speeds). The block-matching process in CINFISOS is done by comparing the most smooth patches from the image in question with patches from other images of known ISO speeds in discrete cosine transformed-space. Experiments are done to validate the effectiveness of the proposed CINFISOS method. The experiments show that by using the proposed CINFISOS method, it can outperform the forgery detections produced by not following the ISO-specific correlation prediction process, in terms of larger area under the ROC curve (AUC-ROC).

7.3 PRNU-based Provenance Inference for Instagram Photos

The PRNU has been extensively studied and found its applications in many practical scenarios in the law-enforcement sector because of its capability of differentiating individual source devices of the same model. However, the emergence of photo-sharing social networking sites (SNS) poses new challenges to the PRNU-based image provenance analysis. In addition to the traditional challenges brought by the vast number of images shared on these sites, the built-in image editing tools on SNSs have exacerbated the issue. One particular problem is that the SNS's built-in image editing tools tend to inflict distortion on images' PRNUs. One well-known example of such a tool is the image filters used by Instagram. In Chapter 5, we observed that some Instagram image filters manipulate the high-frequency bands of the images and hence damage the PRNUs, making source-oriented clustering (SOC) of the filtered images unsatisfactory. The image filters which can significantly distort PRNUs are also identified in Chapter 5.

To address this issue, we propose a three-step clustering framework for Instagram images in Chapter 5. Firstly, the images are separated into two groups according to the filters applied on the images, with Group Malignant (M) containing the filters that significantly distort PRNUs and Group Benign (B) covering the other filters that have no significant impact on PRNUs. The images processed by Group B filters are then clustered and the centroids are extracted from the formed clusters, with each centroid representing the reference PRNU of the corresponding source device. Finally, we use the centroid of each cluster to attract the images processed by Group M filters to complete

the SOC task. To identify the filter applied to each image, a convolutional neural network-based filter-oriented image classifier is also proposed in Chapter 5. To further refine the classification result, by investigating the pairwise correlations and the shared nearest neighbours, images residing in significantly different neighbourhoods measured by these two metrics are excluded from the initial clustering step.

The effectiveness of the proposed three-step clustering framework and the filter-oriented classifier are tested on images from both the VISION dataset [117] and the Dresden Image Dataset [116]. Using an iOS simulator to apply image filters from Instagram on images from the datasets, we run tests on a large number of images. The proposed filter-oriented image classifier is tested on $19,332$ images processed by 18 different filters. It delivers a very promising accuracy of 98.5%. In addition, the proposed clustering framework manages to improve the F1-measure from 47.74% by applying existing clustering methods directly on Instagram images to a much higher F1-measure of 90.33%.

7.4 Detecting Anti-Forensics Attacks on PRNU Using Generative Adversarial Networks

With PRNU-based forensic methods seeing successes in different fields of multimedia forensics, the PRNU becomes a target for anti-forensics attacks. Being a noise-like signal, the PRNU could be attenuated or removed by some simple manipulations like median filtering or Gaussian blurring. When performing PRNU-based forensic methods on a large group of images, filtering out PRNU-absent images can improve the performance and prevent the investigators from making wrong conclusions. Thus, different detection methods for anti-forensics attacks are proposed. Among them, neural network-based classifiers have shown their strength in detecting different types of anti-forensics manipulations. However, the neural network-based classifiers' superior ability to extract manipulation related features could also make them pay too much attention to manipulation-specific features due to the limitation of the training data. As a result, when a neural network-based classifier is used to detect attacks on PRNU from unprecedented manipulations, it may not perform well as the images subject to these unprecedented manipulations may not contain the features the classifier looks for.

To address this problem, in Chapter 6, we propose a generative adversarial networks (GAN) based training strategy for detecting anti-forensic attacks on PRNU. Different to many other GAN-based methods, which aim to generate pristine images to fool the discriminator, the GAN framework proposed in Chapter 6 generates 'lightly attacked' images, which do not possess strong

manipulation-specific features. Also, by generating images from images with PRNU subject to attacks, the generated one could be considered as having their PRNUs attacked as well. Thus, through a GAN framework, the generated images would help the classifier, which is trained as the discriminator in the GAN framework, pay less attention to those manipulation-specific features and focus more on the difference between images with PRNU attacked or not, despite the limitations of the original training set. Experiments on images from the Warwick Image Forensics Dataset show that the proposed GAN-based training strategy can tune the classifier to make it generalise better for detecting unprecedented attacks on PRNUs.

7.5 Future Work

This thesis focuses on studying and addressing the impact from different challenges on PRNU-based image forensics, especially image forgery detection, source oriented image clustering, and detection of anti-forensics attacks. But as digital photography is evolving at the same time, more challenges are coming up. Some possible directions for future researches are as follows.

As mentioned in Chapter 3, computational photography algorithms based on multi-frame merging, e.g., HDR imaging, can pose big threats to PRNU-based image forensics. The image registration step in these multi-frame merging methods usually involves mapping pixels from different locations in different frames taken by the same sensor into one pixel for the final image. Thus, the PRNUs from different locations get mixed up and do not match well with the device's reference PRNU. Furthermore, after multiple frames are merged together, local tone-mapping is usually applied to provide high contrast while keeping the image photo-realistic. This local operation could change the local noise statistics. A big impact would be on PRNU-based image forgery detection. As mentioned in this thesis, a correlation predictor is an important component for many PRNU-based forgery detection methods. To predict the correlations for multi-frame merged images, we have to take the effect of multi-frame alignment into account. In addition, the predictors may have to be used locally to best describe the local noise statistics but how exactly this could be done needs further investigation. The Warwick Image Forensics Dataset provides a good platform to study different multi-frame merging algorithms. Thus, future work could be done based on this dataset.

In addition to multi-frame merging algorithms, multi-camera merging algorithms become popular on consumer-level mobile devices as well. Nowadays, many mobile devices have multiple lenses and each lens has a corresponding sensor, meaning the images from different lenses will have different PRNUs. The lenses on the same device usually have different zoom ranges, providing

different field of views (FOV). Multi-camera merging algorithms are used to merge images from different lenses of the same mobile device to generate images with a user-defined field of view while keeping the image details from multiple lenses. As a result, the merged image will mix the PRNUs from multiple sensors together. Thus, when we perform source camera attribution, including both source camera identification and source oriented image clustering, on these merged images, we have to consider the fact that each of these images contains PRNUs from multiple sensors. With the Warwick Image Forensics Dataset containing images of the same scene from multiple cameras which can be used to merge into a single image, future work for developing source camera attribution methods for multi-camera merged images can be done using the images from the dataset.

In Chapter 4, we proposed an ISO-speed correlation prediction process for PRNU-based image forgery detection methods. The process requires images with similar ISO speeds to the image in question to build the correlation predictor. For future work, more investigations can be carried out to understand how the ISO speed would affect PRNU correlations for JPEG images or images of other compressed formats analytically. Obtaining an analytical model for these images would help to incorporate the ISO speed as a parameter to the correlation predictor. This would make the correlation predictor generalize for all ISO speeds instead of being ISO speed-specific.

In Chapter 5, we investigate and propose a method to mitigate the impact from image filters used by Instagram on PRNU-based source oriented image clustering. Future work can be done by extending the investigation to other social networking sites and developing cross-platform image clustering methods despite the impact of the built-in image editing tools. The social network history of an image can be investigated to reveal which social network sites this image has been uploaded to. This information would help to narrow down the pool of the potential social network sites' built-in image editing tools the image has been applied with. Subsequent forensic investigations could be carried out more precisely.

A generative adversarial networks based training strategy is proposed in Chapter 6 to help neural network-based classifiers perform better on the binary classification of detecting whether an image's PRNU is subject to attack or not. Future work can be done to further improve the classifier's performance by trying different network structures and testing different GAN framework training strategies. In addition, as the detection is done at patch-level, it provides the possibility of using the patch-level detection result for image forgery detection in the future. For example, we can use the binary classification results directly to form the forgery detection map or develop a method which uses the intermediate output of the network to form a confidence

map for the forgery detection, which indicates each pixel's probability of being tampered.

Appendix A

A Case Study on JPEG Compression's Impact on Images of Different ISO Speeds

As different ISO speeds can introduce different levels of noise to the images, such behavior would impact the reference PRNU extraction process as well. A typical method to extract a device's reference PRNU is averaging the noise residuals from flat-field images (images of flattened content, e.g. pure color boards, etc.). The use of flat-field images can mostly avoid the distortion due to image content (e.g. texture, edges, etc.). For a flat-field RAW image, we can approximate its noise model according to Equation (4.1), which means its noise residual consists of both the PRNU and PRNU-irrelevant parts. By averaging the noise residuals of multiple flat-field images from the same device with similar quality of the PRNU, their PRNU-irrelevant part can get attenuated and thus a better approximation of the PRNU can be obtained.

In real-life forensics, the images available for the reference extraction may not be RAW images but in some compressed format, e.g. JPEG images, similar behavior is expected. Also, due to the influence of ISO speed, it is reasonable for us to expect that, with the same number of images, the reference PRNU extracted from lower ISO speed images would be of better quality than the one extracted from images with higher ISO speeds. To verify this, we test the PRNU extracted from varying numbers (from 1 to 50) of flat-field images with different ISO speeds (100, 800 and 6400) from three cameras, namely a Canon 6D MKII, a Nikon D7200 and a Sigma SdQuattro. The images used in this test are JPEG images of a flat color panel and are straight out of the three cameras. To ensure a fair comparison between different ISO speeds, we set the JPEG

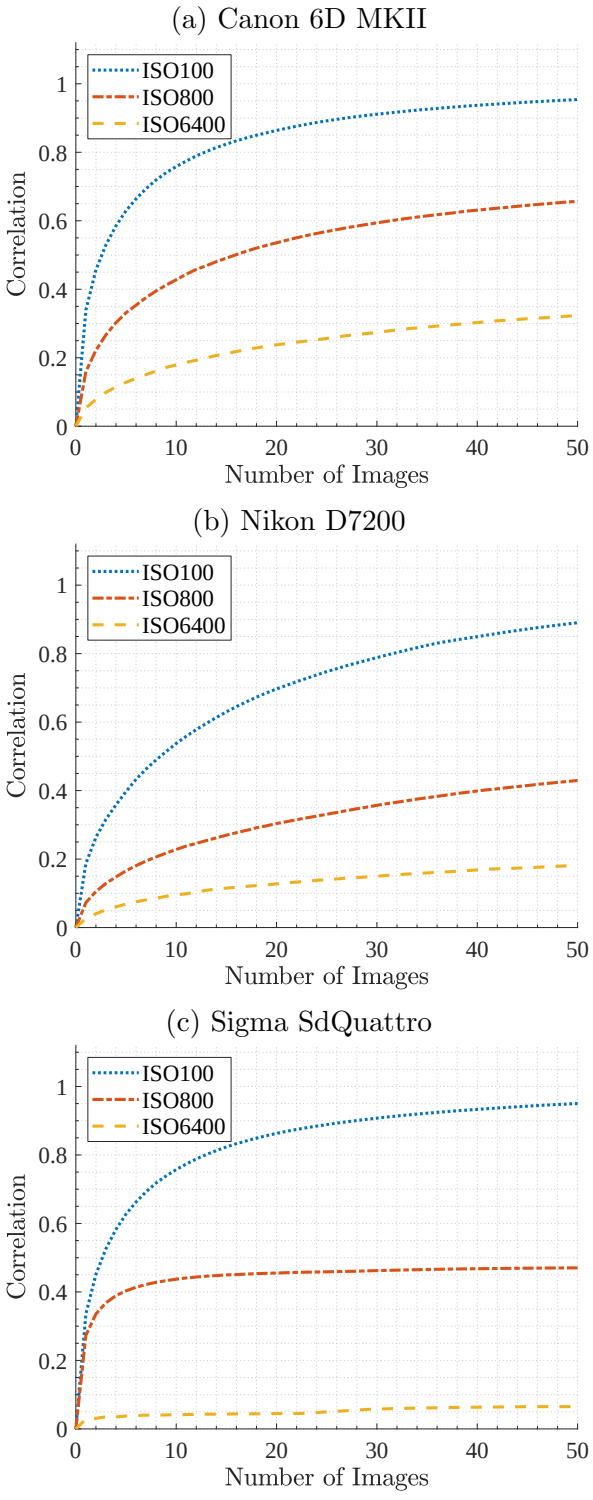


Figure A.1: The plots show how the number of JPEG images used for reference PRNU extraction may affect the quality of the extracted reference PRNU from three cameras: (a) Canon 6D MKII, (b) Nikon D7200 and (c) Sigma SdQuattro. We use the correlation between the extracted reference PRNU with another reference PRNU extracted from 100 flat-field images of ISO speed 100 to indicate the quality of the extracted reference PRNU.

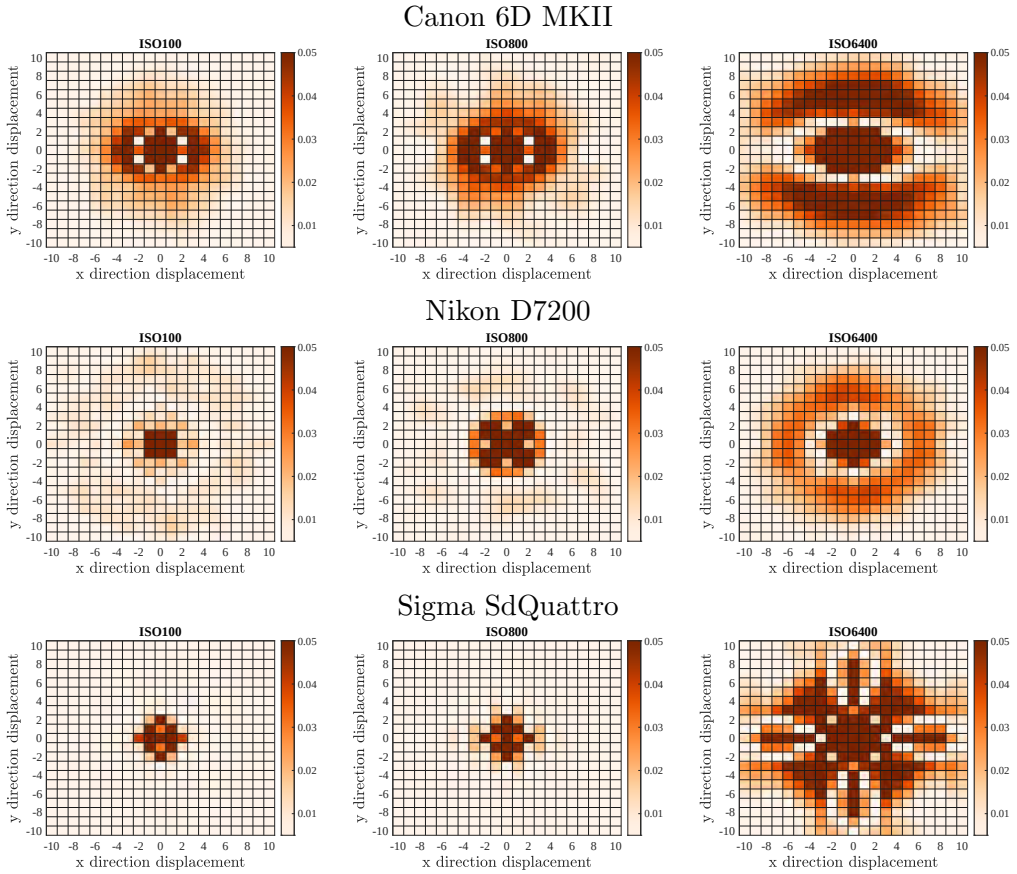


Figure A.2: The auto-correlation of noise residuals from images of different ISO speeds from 3 cameras. Rather than a single peak at $(0, 0)$, auto-correlations have values spread over multiple pixel ranges. As the figure focuses on how far the spreading of auto-correlation reaches, the color bar focus on the range of $[0, 0.05]$. Values bigger than the upper limit 0.05 are also colored in dark brown.

compression quality to the best available setting on each camera for every ISO speed. The quality of the extracted PRNUs is examined by computing the correlation between them and another reference PRNU of the same camera, which in our case is computed from 100 images with ISO speed of 100. We call the PRNUs generated from the one hundred ISO 100 images as the *sample PRNUs*.

In theory, the three sample PRNUs may still differ from the ground truth slightly, the correlation between them and the one extracted from the test images are still representative to tell the difference between the quality of PRNU generated from images of different ISO speeds, as we can see from Fig. A.1. From the figures, we can confirm that the lower ISO speed generates PRNU of better quality. For each ISO speed, the correlation increases as the number of images used to extract the reference PRNU increases.

Furthermore, for different ISO speeds from the same camera, the correlation curves shown in Fig. A.1 tend to converge to different values. It means that no matter how many images are used to extract the reference PRNU, the ones from images of higher ISO speeds can be of worse quality than the ones from a sufficient number of images of lower ISO speeds. Such a phenomenon suggests the incompatibility of PRNU's extracted from higher ISO speed images with the sample PRNU.

We found that this is mainly due to the reason that the PRNU signal remaining in higher ISO images is more prone to be vitiated by low-pass filtering like JPEG compression despite the images are saved under the same JPEG compression quality factor. As the higher ISO speed flat-field images are noisier, they have more PRNU-irrelevant high frequency signals in the image. Thus, when a low-pass filter is applied to them to reduce the amount of high frequency signal remaining in the images to a certain level, the more the PRNU-irrelevant high frequency signals there are in the images, the less information about PRNU would survive under such a compression.

In Fig. A.2, we use the auto-correlations of the flat-field images' noise residual to demonstrate such an effect. For a random noise, as the value of each pixel is independent, its auto-correlation should have a single peak at $(0, 0)$ and is zero elsewhere. However, due to post-processing, especially the JPEG compression, the auto-correlation will spread over multiple pixels and the extend of this spreading can be an indicator of how severe the post-processing may distort the extracted noise residual. From Fig. A.2, for each of the three cameras, we clearly see the trend that as the ISO speed increases, the spreading reaches further. Furthermore, the symmetric spreading shapes as we observed from the plots for the ISO 6400 images, showing the signal spreading is stronger at certain frequency, are more likely to be from JPEG compression which compresses signals of a certain frequency in the images.

Color interpolation (also known as demosaicking) at each pixel involves the colors of the pixels within a neighborhood, which means the color at each pixel does "spread" across a certain neighborhood. Interestingly, unlike the Bayer filter used on the sensors in Canon 6D MKII and Nikon D7200, the Foveon X3 sensor in the Sigma SdQuattro has a stacked color filtering array, which does not require color interpolation. The spreading of the auto-correlation can still be observed with the Sigma SdQuattro. This evidence further justifies that the further spreading of the auto-correlation is more likely to be caused by JPEG compression instead of color interpolation.

Bibliography

[1] Y. Quan and C.-T. Li, "On addressing the impact of ISO speed upon PRNU and forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 190–202, 2021.

[2] Y. Quan, X. Lin, and C.-T. Li, "Provenance analysis for instagram photos," in *Australasian Conference on Data Mining*. Springer, 2018, pp. 372–383.

[3] Y. Quan, C.-T. Li, Y. Zhou, and L. Li, "Warwick image forensics dataset for device fingerprinting in multimedia forensics," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[4] Y. Quan, X. Lin, and C.-T. Li, "Provenance inference for instagram photos through device fingerprinting," *IEEE Access*, vol. 8, pp. 168 309–168 320, 2020.

[5] R. Rouhi, F. Bertini, D. Montesi, X. Lin, Y. Quan, and C.-T. Li, "Hybrid clustering of shared images on social networks for digital forensics," *IEEE Access*, vol. 7, pp. 87 288–87 302, 2019.

[6] M. Utku Celik, G. Sharma, E. Saber, and A. Murat Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 585–595, 2002.

[7] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1593–1601, 2001.

[8] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 3, 1999, pp. 792–796 vol.3.

[9] X. Zhu, A. T. Ho, and P. Marziliano, "A new semi-fragile image watermarking with robust tampering restoration using irregular sampling," *Signal Processing: Image Communication*, vol. 22, no. 5, pp. 515–528, 2007.

[10] C.-C. Chang, P. Tsai, and C.-C. Lin, "Svd-based digital image watermarking scheme," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1577–1586, 2005.

[11] T. K. Tsui, X. Zhang, and D. Androutsos, "Color image watermarking using multidimensional fourier transforms," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 16–28, 2008.

[12] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[13] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[14] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on computing*, vol. 17, no. 2, pp. 281–308, 1988.

[15] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[16] X. Lin, J.-H. Li, S.-L. Wang, A.-W.-C. Liew, F. Cheng, and X.-S. Huang, "Recent advances in passive digital image security forensics: A brief review," *Engineering*, vol. 4, no. 1, pp. 29 – 39, 2018, cybersecurity.

[17] S. K. Choi, E. Y. Lam, and K. K.-Y. Wong, "Source camera identification using footprints from lens aberration," in *Digital Photography II*, vol. 6069. International Society for Optics and Photonics, 2006, p. 60690J.

[18] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proceedings of the 8th Workshop on Multimedia and Security*. ACM, 2006, pp. 48–55.

[19] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 250–263, 2014.

[20] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[21] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072. International Society for Optics and Photonics, 2006, p. 60720Y.

[22] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, 2010.

[23] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[24] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF approach for PRNU-based image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, Apr 2014.

[25] C.-T. Li and X. Lin, "A fast source-oriented image clustering method for digital forensics," *EURASIP Journal on Image and Video Processing: Special Issues on Image and Video Forensics for Social Media analysis*, vol. 1, pp. 69–84, Oct. 2017.

[26] X. Lin and C.-T. Li, "Large-scale image clustering based on camera fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 793–808, Apr 2017.

[27] S. Bayram, H. T. Sencar, and N. Memon, "Classification of digital camera-models based on demosaicing artifacts," *Digital Investigation*, vol. 5, no. 1-2, pp. 49–59, 2008.

[28] S. Gao, G. Xu, and R.-M. Hu, "Camera model identification based on the characteristic of cfa and interpolation," in *Digital Forensics and Watermarking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 268–280.

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

[30] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[31] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1497–1500.

[32] A. Swaminathan, M. Wu, and K. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 91–106, 2007.

[33] Z. Lin, R. Wang, X. Tang, and H. Shum, "Detecting doctored images using camera response normality and consistency," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun 2005, pp. 1087–1092 vol. 1.

[34] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 549–552.

[35] G. Cao, Y. Zhao, and R. Ni, "Forensic estimation of gamma correction in digital images," in *2010 IEEE International Conference on Image Processing*, 2010, pp. 2097–2100.

[36] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Digital Forensic Research Workshop*, 2003, pp. 5–8.

[37] D. Fu, Y. Q. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505. International Society for Optics and Photonics, 2007, p. 65051L.

[38] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492 – 2501, 2009.

[39] F. Huang, J. Huang, and Y. Q. Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 848–856, 2010.

[40] J. Yang, J. Xie, G. Zhu, S. Kwong, and Y. Shi, "An effective method for detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1933–1942, 2014.

[41] X. Huang, S. Wang, and G. Liu, "Detecting double JPEG compression with same quantization matrix based on dense cnn feature," in *2018 25th*

IEEE International Conference on Image Processing (ICIP), 2018, pp. 3813–3817.

[42] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 2. IEEE, 2007, pp. II–217.

[43] Z. Qu, W. Luo, and J. Huang, "A convolutive mixing model for shifted double JPEG compression with application to passive image authentication," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 1661–1664.

[44] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2012.

[45] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[46] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, 2017.

[47] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato, "A classification engine for image ballistics of social data," in *Image Analysis and Processing - ICIAP 2017*. Springer International Publishing, 2017, pp. 625–636.

[48] I. Amerini, C.-T. Li, and R. Caldelli, "Social network identification through image classification with cnn," *IEEE access*, vol. 7, pp. 35 264–35 273, 2019.

[49] Q. Phan, G. Boato, R. Caldelli, and I. Amerini, "Tracking multiple image sharing on social networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8266–8270.

[50] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *Media forensics and security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 754110.

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

[52] H. Yuan, "Blind forensics of median filtering in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1335–1345, 2011.

[53] C. Chen and J. Ni, "Median filtering detection using edge based prediction matrix," in *Digital Forensics and Watermarking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 361–375.

[54] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

[55] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain based approach," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4699–4710, 2013.

[56] Y. Zhang, S. Li, S. Wang, and Y. Q. Shi, "Revealing the traces of median filtering using high-order local ternary patterns," *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 275–279, 2014.

[57] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, 2015.

[58] F. Ding, G. Zhu, J. Yang, J. Xie, and Y. Shi, "Edge perpendicular binary coding for usm sharpening detection," *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 327–331, 2015.

[59] G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 603–606, 2011.

[60] F. Ding, G. Zhu, and Y. Q. Shi, "A novel method for detecting image sharpening based on local binary pattern," in *Digital-Forensics and Watermarking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 180–191.

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

[62] M. Kirchner and T. Gloe, "On resampling detection in re-compressed images," in *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, 2009, pp. 21–25.

[63] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM workshop on Multimedia and security*, 2008, pp. 11–20.

[64] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Technical Report TR2004-515, Department of Computer Science, Dartmouth College*, pp. 1–11, 2004.

[65] J. Fridrich, B. D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*, 2003.

[66] S. Bayram, H. Taha Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1053–1056.

[67] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–18, 2012.

[68] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with j-linkage," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 659 – 669, 2013.

[69] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, 2010.

[70] M. C. Stamm and K. J. R. Liu, "Forensic estimation and reconstruction of a contrast enhancement mapping," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 1698–1701.

[71] X. Lin, C. Li, and Y. Hu, "Exposing image forgery through the detection of contrast enhancement," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 4467–4471.

[72] X. Lin, X. Wei, and C.-T. Li, "Two improved forensic methods of detecting contrast enhancement in digital images," in *Media Watermarking, Security, and Forensics 2014*, vol. 9028. International Society for Optics and Photonics, 2014, p. 90280X.

[73] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proceedings of the 7th workshop on Multimedia and security*, 2005, pp. 1–10.

[74] M. K. Johnson and H. Farid, "Exposing digital forgeries through specular highlights on the eye," in *Information Hiding*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 311–325.

[75] E. Kee, J. F. O'brien, and H. Farid, "Exposing photo manipulation from shading and shadows," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 5, pp. 1–21, 2014.

[76] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, 2007.

[77] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, ser. IH and MMSec '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 5–10.

[78] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 159–164.

[79] J. Yu, Y. Zhan, J. Yang, and X. Kang, "A multi-purpose image counter-anti-forensic method using convolutional neural networks," in *15th International Workshop on Digital Forensics and Watermarking, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers*, vol. 10082. Springer, 2017, p. 3.

[80] T.-D. Lee and C.-N. Yang, "Statistical theory of equations of state and phase transitions. II. lattice gas and Ising model," *Physical Review*, vol. 87, no. 3, p. 410, 1952.

[81] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM journal on applied mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[82] P. Combettes and J.-C. Pesquet, "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators," *Set-Valued and variational analysis*, vol. 20, no. 2, pp. 307–330, 2012.

[83] G. Chierchia, S. Parrilli, G. Poggi, L. Verdoliva, and C. Sansone, "PRNU-based detection of small-size image forgeries," in *2011 17th International Conference on Digital Signal Processing (DSP)*, 2011, pp. 1–6.

[84] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6231–6235.

[85] W. Zhang, X. Tang, Z. Yang, and S. Niu, "Multi-scale segmentation strategies in PRNU-based image tampering localization," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 20 113–20 132, 2019.

[86] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, 2016.

[87] P. Korus and J. Huang, "Multi-scale fusion for improved localization of malicious tampering in digital images," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1312–1326, 2016.

[88] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008.

[89] C.-T. Li, "Unsupervised classification of digital images using enhanced sensor pattern noise," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 3429–3432.

[90] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proceedings of IEEE International Workshop on Information Forensics and Security*, Dec 2010, pp. 1–5.

[91] L. J. G. Villalba, A. L. S. Orozco, and J. R. Corripio, "Smartphone image clustering," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1927–1940, 2015.

[92] B. Liu, H. Lee, Y. Hu, and C. Choi, "On classification of source cameras: A graph based approach," in *2010 IEEE International Workshop on Information Forensics and Security*, Dec 2010, pp. 1–5.

[93] S.X. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 313–319 vol.1.

[94] S. Luan, X. Kong, B. Wang, Y. Guo, and X. You, "Silhouette coefficient based approach on cell-phone classification for unknown source images," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 6744–6747.

[95] I. Amerini, R. Caldelli, P. Crescenzi, A. Del Mastio, and A. Marino, "Blind image clustering based on the normalized cuts criterion for camera identification," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 831–843, 2014.

[96] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[97] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2197–2211, 2017.

[98] Q. Phan, G. Boato, and F. G. B. De Natale, "Accurate and scalable image clustering based on sparse representation of camera fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1902–1916, 2019.

[99] S. Georgievska, R. Bakhshi, A. Gavai, A. Sclocco, and B. van Werkhoven, "Clustering image noise patterns by embedding and visualization for common source camera detection," *Digital Investigation*, vol. 23, pp. 22–30, 2017.

[100] S. Khan and T. Bianchi, "Fast image clustering based on camera fingerprint ordering," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 766–771.

[101] X. Lin and C.-T. Li, "Rotation-invariant binary representation of sensor pattern noise for source-oriented image and video clustering," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.

[102] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819. International Society for Optics and Photonics, 2008, p. 68190E.

[103] M. Goljan and J. Fridrich, "Sensor fingerprint digests for fast camera identification from geometrically distorted images," in *Media Watermarking, Security, and Forensics 2013*, vol. 8665. International Society for Optics and Photonics, 2013, p. 86650B.

[104] S. Bayram, H. T. Sencar, and N. D. Memon, "Seam-carving based anonymization against image video source attribution," in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 272–277.

[105] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 10–es.

[106] S. Taspinar, M. Mohanty, and N. Memon, "PRNU-Based camera attribution from multiple seam-carved images," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3065–3080, 2017.

[107] A. E. Dirik, H. T. Sencar, and N. Memon, "Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2277–2290, 2014.

[108] P. Sengupta, V. U. Sameer, R. Naskar, and E. Kalaimannan, "Source anonymization of digital images: A counter–forensic attack on PRNU based source identification techniques," in *Proceedings of the Conference on Digital Forensics, Security and Law*. Association of Digital Forensics, Security and Law, 2017, pp. 95–105.

[109] L. J. G. Villalba, A. L. S. Orozco, J. R. Corripio, and J. Hernandez-Castro, "A PRNU-based counter-forensic method to manipulate smartphone image source identification techniques," *Future Generation Computer Systems*, vol. 76, pp. 418–427, 2017.

[110] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1050–1065, 2011.

[111] M. Wang, Z. Chen, W. Fan, and Z. Xiong, "Countering anti-forensics to wavelet-based compression," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5382–5386.

[112] H. Cao and A. C. Kot, "Manipulation detection on image patches using fusionboost," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 992–1002, 2012.

[113] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 149–154.

[114] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[115] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 31–45, 2018.

[116] T. Gloe and R. Böhme, "The 'Dresden Image Database' for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

[117] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, Oct 2017.

[118] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[119] L. Lin, W. Chen, Y. Wang, S. Reinder, Y. Guan, J. Newman, and M. Wu, "The impact of exposure settings in digital image forensics," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 540–544.

[120] S. Mann and R. Picard, "Being 'undigital' with digital cameras," *MIT Media Lab Perceptual*, vol. 1, p. 2, 1994.

[121] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *ACM SIGGRAPH 2008 classes*. ACM, 2008, p. 31.

[122] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 192, 2016.

[123] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep CNNs," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 178, 2017.

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

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

[126] C. Galdi, F. Hartung, and J.-L. Dugelay, "Socrates: A database of realistic data for source camera recognition on smartphones," in *Proceedings of International Conference Pattern Recognition Applications and Methods*, 2019, pp. 19–21.

[127] H. Tian, Y. Xiao, G. Cao, Y. Zhang, Z. Xu, and Y. Zhao, "Daxing smartphone identification dataset," *IEEE Access*, vol. 7, pp. 101 046–101 053, 2019.

[128] O. Shaya, P. Yang, Y. Ni, R.and Zhao, and A. Piva, "A new dataset for source identification of high dynamic range images," *Sensors*, vol. 18, no. 11, p. 3801, 2018.

[129] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

[130] X. Lin and C.-T. Li, "Preprocessing reference sensor pattern noise via spectrum equalization," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 11, no. 1, pp. 126–140, 2016.

[131] W. van Houten and Z. Geradts, "Using anisotropic diffusion for efficient extraction of sensor noise in camera identification," *Journal of Forensic Sciences*, vol. 57, no. 2, pp. 521–527, 2012.

[132] A. Cooper, "Improved photo response non-uniformity (PRNU) based source camera identification," *Forensic Science International*, vol. 226, no. 1-3, pp. 132–141, 2013.

[133] F. Gisolf, A. Malgoezar, T. Baar, and Z. Geradts, "Improving source camera identification using a simplified total variation based noise removal algorithm," *Digital Investigation*, vol. 10, no. 3, pp. 207–214, 2013.

[134] H. Zeng and X. Kang, "Fast source camera identification using content adaptive guided image filter," *Journal of Forensic Sciences*, vol. 61, no. 2, pp. 520–526, 2016.

[135] C.-T. Li and Y. Li, "Color-decoupled photo response non-uniformity for digital image forensics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 260–271, Feb 2012.

[136] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, Mar 1994.

[137] A. Foi, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing*, vol. 89, no. 12, pp. 2609–2629, 2009.

[138] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2006, pp. 901–908.

[139] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, Dec 2013.

[140] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 2209–2216.

[141] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1683–1691.

[142] D. Cozzolino, F. Marra, D. Gragnaniello, G. Poggi, and L. Verdoliva, "Combining PRNU and noiseprint for robust and efficient device source identification," *EURASIP Journal on Information Security*, vol. 2020, no. 1, pp. 1–12, 2020.

[143] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media forensics and security*, vol. 7254. International Society for Optics and Photonics, 2009, p. 72540I.

[144] R. Satta and P. Stirparo, "On the usage of sensor pattern noise for picture-to-identity linking through social network accounts," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, Jan 2014, pp. 5–11.

[145] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, June 2017.

[146] I. Amerini, T. Uricchio, and R. Caldelli, "Tracing images back to their social network of origin: A cnn-based approach," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[147] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on computers*, vol. 100, no. 11, pp. 1025–1034, 1973.

[148] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[149] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv*, 2015.

[150] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[151] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[152] D. Kim, H. Jang, S. Mun, S. Choi, and H. Lee, "Median filtered image restoration and anti-forensics using adversarial networks," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 278–282, 2018.

[153] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.