DIGITAL IMAGE FORENSICS IN THE CONTEXT OF BIOMETRICS

By

Sudipta Banerjee

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2020

ABSTRACT

DIGITAL IMAGE FORENSICS IN THE CONTEXT OF BIOMETRICS

By

Sudipta Banerjee

Digital image forensics entails the deduction of the origin, history and authenticity of a digital image. While a number of powerful techniques have been developed for this purpose, much of the focus has been on images depicting natural scenes and generic objects. In this thesis, we direct our focus on biometric images, viz., iris, ocular and face images.

Firstly, we assess the viability of using existing sensor identification schemes developed for visible spectrum images on near-infrared (NIR) iris and ocular images. These schemes are based on estimating the multiplicative sensor noise that is embedded in an input image. Further, we conduct a study analyzing the impact of photometric modifications on the robustness of the schemes. Secondly, we develop a method for sensor de-identificaton, where the sensor noise in an image is suppressed but its biometric utility is retained. This enhances privacy by unlinking an image from its camera sensor and, subsequently, the owner of the camera. Thirdly, we develop methods for constructing an image phylogeny tree from a set of near-duplicate images. An image phylogeny tree captures the relationship between subtly modified images by computing a directed acyclic graph that depicts the sequence in which the images were modified. Our primary contribution in this regard is the use of complex basis functions to model any arbitrary transformation between a pair of images and the design of a likelihood ratio based framework for determining the original and modified image in the pair. We are currently integrating a graph-based deep learning approach with sensor-specific information to refine and improve the performance of the proposed image phylogeny algorithm.

Dedicated to Maa and Baba

ACKNOWLEDGMENTS

I wish to express my sincerest gratitude to my doctoral advisor and mentor, Professor Arun Ross, for providing me with outstanding resources to build my research appetite and grooming me as a graduate student. His expertise, unequivocal presentation skills, and a quest for excellence have motivated me to pursue quality in all my tasks. He has taught me the importance of seeking answers to three questions in research - *what, why* and *how*. It is necessary but not sufficient to just know what works, but also why does a specific technique work the way it does, and how does it work, are equally important questions. He has always encouraged me to strive for holistic experience as a doctoral candidate by giving me opportunities to attend conferences, workshops, summer school, and volunteering in outreach programs. For that, I am grateful to him. I am also thankful to Professor Anil Kumar Jain, whom I am fortunate to have on my doctoral committee. He has been a great source of inspiration and encouragement to me. I will take this opportunity to thank Professor Selin Aviyente and Professor Yiying Tong to serve on my doctoral committee and providing me with their expert advice in my research. I appreciate the support given by Professor Sandeep Kulkarni, the Associate Chair of the Graduate Studies, and Professors Xiaoming Liu and Vishnu Boddeti. I am also thankful to Professor Ananda S. Chowdhury, my Masters' adviser who encouraged me to pursue doctoral studies at Michigan State University.

I feel blessed to have very supportive research lab mates and colleagues who have been extremely patient with me and supported me through this journey of five and a half years. They helped me, critiqued me, and prepared me to defend my research. Also, they celebrated with me when I would receive an award, and consoled me when my paper would get rejected. They have been critical in my journey which is inching towards its end now - (in the order I encountered them) Thomas Swearingen, Steven Hoffman, Dr. Denton (Denny) Bobeldyk, Dr. Yaohui (Eric) Ding, Raghunandan Pasula, Aaron Gonzalez, Melissa Dale and Jessie, Anurag Chowdhury, Dr. Vahid Mirjalili, Renu Sharma, Achsah Ledala, Shivangi Yadav, Dr. Darshika Jauhari and Austin Cozzo. I want to thank Kelly Climer, Brenda Hodge, and Erin Dunlop for their support and help.

Personally, none of this would have been possible without God's grace, and I could feel the blessing and unconditional love and support showered upon me through my parents. My family has been a pillar of support for me right from the beginning when I decided to come abroad for the doctoral program. They have continued to motivate me whenever I would doubt myself. My brother and sister-in-law expressed how proud they are of me, something that I will always cherish.

Friends, what would I do without them? I always doubted about making new friends in the States. I always thought I would stop making good friends after school and college. Pratiti, Sangita, and Monalisa have been great friends, and they will be my friends for life. I am grateful to have added more to the list of people, my extended support line. They have been by my side whenever I needed help, and I hope to continue our friendship even after we part ways from Michigan State University. A very special thanks to Inci, Nilay and Priyanka, and Vidhya and Sai - you guys mean a lot to me, and thank you for having me as your friend!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Biometrics

Biometrics refers to the science of recognizing individuals based on their physical or behavioral attributes [92]. Examples of these attributes include face, fingerprint, iris, voice, gait, hand geometry, keystroke dynamics and signature [90]. Such attributes are typically intrinsic to a particular individual. Therefore, biometrics can be used for identification and verification purposes. Identification pertains to recognizing an individual from a set of several individuals. Verification, on the other hand, involves confirming the identity claimed by an individual. Biometrics has found its way into our daily lives, be it in the form of access control such as TouchID or FaceID on the iPhones, or speaker recognition on Alexa. The scope of applications of biometrics has extended beyond the original utility of authentication, and is slowly gaining momentum towards ancillary motives, such as gender prediction from ocular images [34], demographic prediction from face images [82], etc. This demonstrates the capability of biometrics to cast a strong influence in different disciplines such as medicine, online marketing, social media, and much more.

A biometric system has two modes of operation — an enrollment stage and a verification stage [90]. During the enrollment stage, a gallery is constructed by acquiring biometric data from a large number of individuals. The biometric template computed from the biometric data serves as a personal identifier for each enrollee in the gallery. During the verification stage, a user presents their biometric template for authentication. The probe or the query sample can then be used for either identification or verification. The entire pipeline can be succinctly described using the following steps:

- **Acquisition:** The *sensor* is responsible for the acquisition of data from an individual. Images of faces, fingerprints and irides are acquired using special sensors developed for the respective

modality. Iris sensors typically operate in the near-infrared (NIR) spectrum compared to face sensors that operate in the visible (VIS) spectrum. The application scenario also dictates the type of sensor used. For example, covert operations may use thermal cameras operating in a night-time environment. Speaker recognition requires microphones to acquire the audio data. Occasionally, the sensor module may also apply some form of on-board processing, such as red-eye correction or converting the data into a format suitable for efficient storage.

- **Extraction:** The *feature extraction* module processes the data acquired by the sensor and distills a compact representation that encapsulates the most relevant components in the original data. Minutia are the feature descriptors for a fingerprint image, whereas, a binary IrisCode is the distilled feature representation for an iris image.

- **Comparison:** The *matcher* is the final component that accepts two features extracted from two sets of data and compares them to compute a match score. The match score can be in the form of a similarity score or a distance score. This score measures the degree of disparity between the two feature representations, which in turn, can be used to infer whether two representations belong to the same identity or not.

1.2 Digital Image Forensics

Digital image forensics refers to the science of integrity verification of images and videos [65, 134]. The history of image tampering dates back to at least a hundred years ago, when there were accounts of images being modified by inserting or eliminating an element present in the image. The element can be an object, or a person. The concern is still the same in current times, but now involves a wide gamut of possibilities in addition to insertion and deletion operations. Image editing operations include moving objects around in an image, adding an emotional expression to a face image, adding special effects, cosmetic enhancements, and even synthesizing an entire image. Digital image forensics involves the development and study of methods that can be employed to

indicate whether an image has been tampered with or not, and identify the tampered region within the image [137]. The forensics can be conducted at different levels as indicated in [66]:

- *Format-Based Analysis:* The format of the file in which the acquired image is stored can provide some useful cues as to image tampering. JPEG headers contain information about the compression rate and can be used to indicate whether an image has undergone double compression. This happens when an original image is modified successively and is saved in the form of compressed files. The JPEG header contains an image's EXIF (Exchangeable Image File Format) metadata. The metadata is organized in the form of image file directories (IFDs), which may include the thumbnail, the GPS location and some additional details. The EXIF metadata can serve as a unique camera signature. This signature can be compared with a gallery of known camera signatures to assess whether a particular image has undergone any tampering, as tampering will cause a mismatch in the metadata.

- *Camera-Based Analysis:* Most digital cameras are equipped with a single CCD or a CMOS sensor integrated with color filter array (CFA) [134]. Bayer array is the most widely used CFA in a majority of cameras. Each CFA records a single value of a color channel (red, green or blue) at a single pixel location. This implies that the other two color channel values at a particular pixel have to be interpolated using the color values present in neighboring pixels. This estimation is referred to as demosaicking. The interpolation may inadvertently introduce correlations which are typically of periodic nature due to the fact that the CFA is arranged in a periodic fashion. Image tampering, such as splicing of two images to produce a composite image, destroys the correlations or introduces inconsistencies in correlations of the CFA array. These inconsistencies or lack of correlations can therefore be leveraged to identify doctored images. Another phenomenon intrinsic to the camera's optical imaging system is known as chromatic aberration that can be harnessed to expose image forgery. As the name suggests, this is an error arising due to the failure of the lens to focus light of all wavelengths at a single point. This results in spitting of the light beam leading

in a polychromatic light. Chromatic aberration leads to color imperfections that appear consistently across the image, but tampering with a particular region of an image alters the magnitude and direction of the chromatic distortion, and therefore, can be subsequently used for indicating tampering. Finally, the recently proposed sensor pattern noise, arising due to anomalies in the fabrication process has been successfully used for image forensic purposes. This noise component persists in each image acquired using a camera over time and serves as a unique 'camera fingerprint'. Details about the camera sensor pattern noise based image forensics will be discussed further in the following section.

- *Pixel-Based Analysis:* Insertion, deletion or cloning of objects in an image are usually accompanied by re-sampling operations to make the forged image look realistic. Otherwise, the image may look irregular or abnormal. This involves geometric registration and possibly modification of the pixel intensity values for the foreground to blend well into the background of the composite. Determining the geometry and color of the cloned region typically involves keypoint-based matching, where keypoints indicate salient regions in an image. Scale-Invariant Feature Transform (SIFT) can be used to extract the keypoints which are then matched using an Euclidean norm in an image. The largest set of matching keypoints corresponds to a possible cloned region in that image.

- *Statistical Modeling-Based Analysis:* Projection of data onto a linear subspace can be used for discriminating between different classes of images, say between original images and computer rendered images. Images generated using computer graphics are typically rendered under ideal assumptions pertaining to the geometry and optical models, which is not the case in the case of real photographic images acquired using cameras. The deviations in the underlying statistical distributions between the synthesized images and real images will result in disparate projections onto the subspace. These distinct projections can then be used to classify the synthetic images.

- *Geometric Modeling-Based Analysis:* The imaging model provides the perspective projection

of a point in a 3-D world coordinate system to a 2-D homogeneous coordinate system. Image-splicing operations can introduce geometric distortions that can be estimated in terms of transformation matrix coefficients. The differences in the estimated parameters (matrix coefficients) can be used for detecting presence of image tampering. Other factors such as geometric inconsistencies arising due to fake reflections or shadows in altered images can be used as cues to indicate presence of image tampering.

- *Physics-Based Analysis:* The direction of the light source can also be used as an additional cue into detection of image tampering. The 2-D lighting model used in images typically assumes a Lambertian reflecting surface. Inconsistent lighting directions can suggest, but does not conclusively indicate, presence of tampering. The properties that hold good for a Lambertian surface can be used for analyzing the physical discrepancies in an image to indicate manipulations.

1.3 Image Forensics in the Context of Biometrics

The field of *digital image forensics* uses scientific principles to establish the origin and authenticity of digital images [66]. The proliferation of digital images in a number of applications, ranging from social media [40] to law enforcement [120, 138], has further accentuated the need to develop effective image forensic tools for a myriad of purposes. As described in Section 1.1, a biometric system uses a *sensor* module for acquiring the biometric data, usually in the form of an image. A biometric image, therefore, contains traces of both sensor-specific and biometric-specific information. As discussed in Section 1.2, camera-based analysis and pixel-based analysis are two image forensic schemes. The camera-based analysis of a biometric image can be used for biometric sensor identification - this comes under sensor-based forensics. The pixel-level analysis of a biometric image can be used for image modification detection - this comes under content-based analysis. We encompass both perspectives of sensor-based and content-based analyses to gain a better understanding of image forensics in the context of biometric images (see Figure 1.1) in our work.

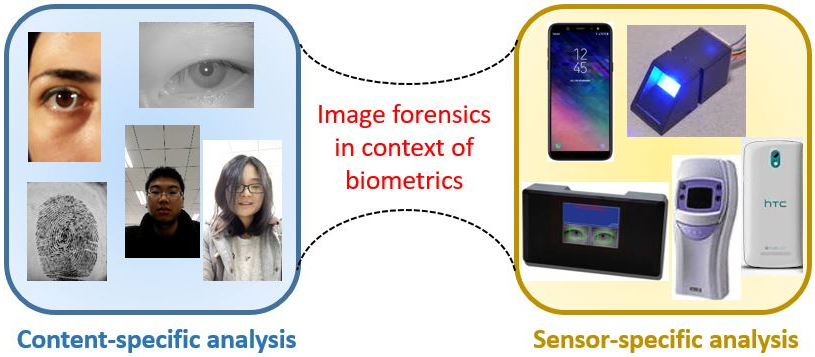
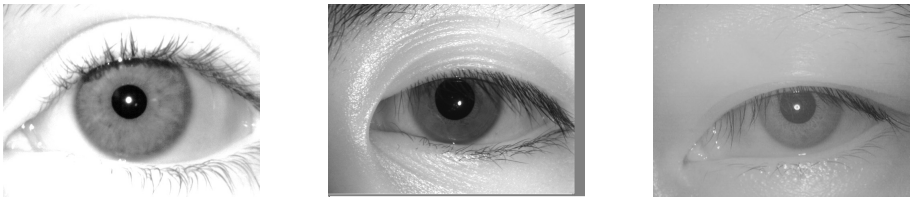


Figure 1.1: The overarching objective in this thesis is the integration of content-specific and sensor-specific characteristics present in a biometric image to develop image forensic strategies in the context of biometrics.



(a) (b) (c)

Figure 1.2: Examples of Near-Infrared (NIR) iris images captured using (a) Aoptix Insight, (b) LG iCAM4000 and (c) IrisCam V2 sensors.

1.3.1 Sensor-based forensics

Although we will explore sensor-forensics for different types of biometric modalities, we begin our research in the context of iris modality. This is due to that fact that limited work has been done on ocular images [154]. Most iris recognition systems acquire a near infrared (NIR) image of the iris[1] region (Figure 1.2). Recent work based on digital image forensics has established the possibility of deducing sensor information from the iris image alone [18, 19, 52, 71, 154]. Determining sensor information (e.g., brand name of sensor) from the iris image has several benefits, especially in the context of digital image forensics.

Validating Metadata: Iris datasets usually have some metadata associated with them such as

[1]It must be noted that a typical iris sensor captures the *ocular* region extending beyond the *iris*. The term *iris image* has been used interchangeably with the term *ocular image* captured by the sensor.

the date and time when the images were taken or modified, details of the camera used for image acquisition, data collection protocol employed, etc. However, this metadata may be inadvertently or deliberately corrupted. Image forensics can be used to verify if the images in a dataset did indeed originate from the claimed source.

Sensor-specific Processing: Some sensors may perform on-board processing of the acquired iris image. Knowledge of the sensor may then be used to recover the pre-processing history of an iris image. In some cases, sensor information can be used to photometrically or geometrically adjust an iris image. This can also be used to facilitate sensor interoperability.

Device Validation: In biometric applications such as banking, it may be necessary to authenticate both the subject and the device itself. In such cases, deducing device information from the biometric image will be useful [71].

Forensic Applications: Linking multiple iris images to a particular sensor source may be required in forensic applications in order to validate the source of the iris images and to secure the chain of custody.

Tampered Images: Sensor identification schemes can be used to detect iris images that have been tampered with. For example, if the pixel artifacts observed in an image are not consistent with the sensor from which it was claimed to be obtained, there is a possibility that it has been modified after acquisition.

The literature on image forensics, which includes deducing sensor information from digital images, is rapidly developing. Early work focused on extracting information about dead pixels [74] and color filter array interpolation artifacts [27] of a sensor from its images. Recent work has shown that Sensor Pattern Noise (SPN) extracted from images can be used to recognize the acquisition device [112, 113]. The two primary sources of noise in the image acquisition stage are shot noise and pattern noise [41]. Shot noise, also known as photonic noise, is a random component. Pattern noise, on the other hand, is deterministic in nature and remains in every image that has been captured by a given sensor and can, therefore, be used to identify the sensor model. The two primary components of pattern noise are Fixed Pattern Noise (FPN) and Photo Response Non

Uniformity (PRNU) [41]. FPN is generated by dark currents. It is an additive component and can be suppressed by flat fielding where a dark frame is subtracted from every image taken by the camera. On the other hand, PRNU is the more dominant multiplicative term arising as a consequence of minor defects in the sensor manufacturing process. PRNU arises due to variation in sensitivity of individual pixels to the same light intensity. Extensive work has been done in this regard to reliably estimate PRNU [41, 106, 108, 109]. But these methods have been developed specifically for sensors operating in the visible (VIS) spectrum.

Iris sensors, on the other hand, primarily operate in the NIR spectrum. This poses a new challenge to traditional image forensics. For example, NIR focal plane arrays have larger pixel size compared to the CCD arrays employed by VIS cameras [118]. Furthermore, in some cases, the materials used for manufacturing NIR detectors can be different from those used in VIS cameras [158]. These factors can impact the PRNU estimation process. Therefore, it is necessary to determine if PRNU estimation schemes developed for VIS cameras can be appropriated to NIR sensors.

There has been limited work on sensor identification in the context of NIR iris images by Uhl and Höller [154], Kalka *et al.* [98], and Debiasi and Uhl [53].

1.3.1.1 Sensor Identification Methods

The general framework for sensor identification from an input iris image is summarized in Figure 1.3. The crux of the framework lies in estimating the sensor reference pattern from a set of training images emerging from the sensor. This reference pattern is then correlated with the noise residual pattern extracted from an input iris image in order to compute a correlation score. Given multiple reference patterns corresponding to different sensors, the test iris image is assigned to that sensor class whose reference pattern results in the highest correlation score.

If an imaging sensor is illuminated with a uniform light intensity $\boldsymbol{Y}$, the sensor registers the signal as, $\boldsymbol{I} \approx (1 + \boldsymbol{K}) \times \boldsymbol{Y}$ [41]. The multiplicative term ($\boldsymbol{K}$) present in the output $\boldsymbol{I}$ signal is the PRNU corresponding to the sensor used to acquire the image. The term $\boldsymbol{K}$ is intrinsic to each

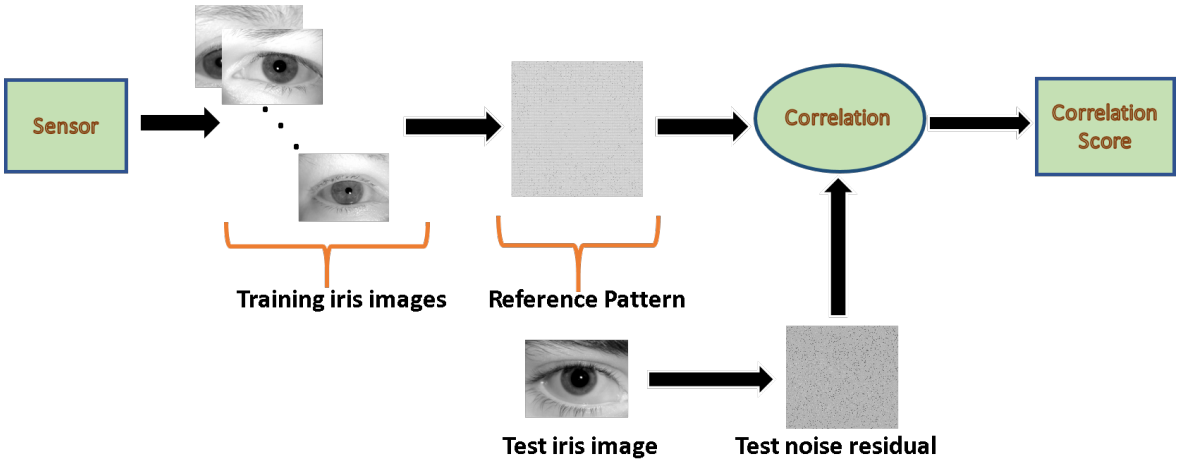


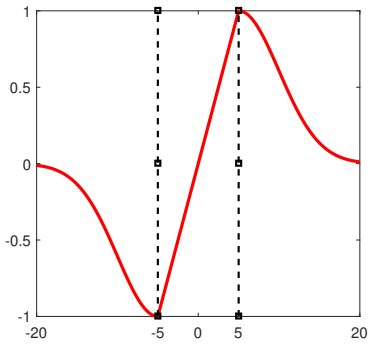
Figure 1.3: General framework for sensor identification from an iris image.

sensor, so it is also referred as the sensor reference pattern. Next, we describe methods used to estimate the sensor reference pattern and perform PRNU-based sensor identification.

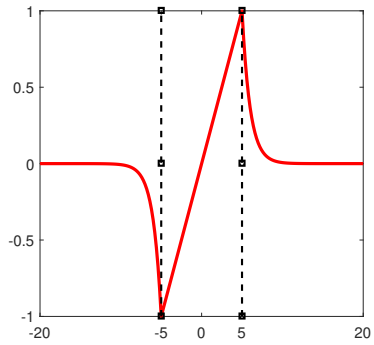
Basic SPN: Lukas *et al.* [112] proposed extraction of the PRNU by denoising the original image using 8-tap Daubechies Quadrature Mirror filter. The camera reference pattern **K**, which is a matrix of same dimensions as the sensor, is constructed using the average of N noise residuals, W_i, $i = 1 \ldots N$, corresponding to N training images. The noise residual corresponding to i^{th} training image I_i is calculated as $W_i = I_i - F(I_i)$ and the reference pattern is computed as $\boldsymbol{K} = \frac{\sum_{i=1}^{N} W_i}{N}$. Here, F represents the wavelet based denoising function as used in [112].

MLE SPN: Chen *et al.* [41] used Maximum Likelihood Estimation (MLE) to obtain a more accurate estimate of PRNU. The authors also employed zero-mean operation and Wiener filtering to reduce the interpolation artifacts. The interpolation artifacts stem from the Bayer pattern and should, therefore, not impact NIR images because infrared sensors do not use color filter arrays (CFA). The MLE camera reference pattern is computed as $\mathbf{K} = \frac{\sum_{i=1}^{N} W_i I_i}{\sum_{i=1}^{N} I_i I_i}$. The MLE noise residual for a given test image Y is computed as $t_Y = \mathbf{K}Y$.

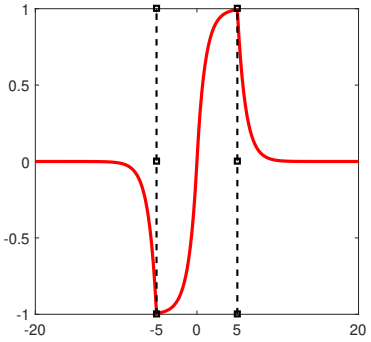
Enhanced SPN: Li [106] proposed an enhancement scheme to attenuate the scene details, which can contaminate the noise residual obtained using Basic SPN. The author proposed five enhancement models to subdue the scene influences by modulating the magnitude of the noise components in the wavelet domain (Figure 1.4). Only the test noise residuals are enhanced, because



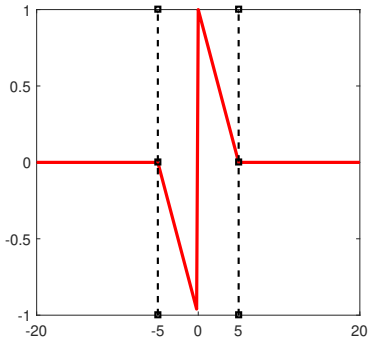
(a) Model I



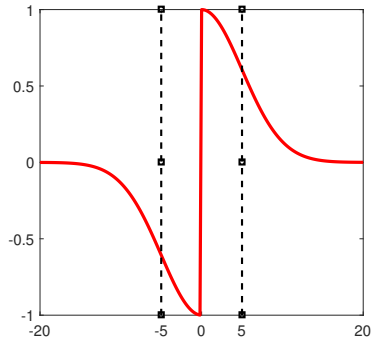
(b) Model II



(c) Model III



(d) Model IV



(e) Model V

Figure 1.4: Photo Response Non-Uniformity (PRNU) enhancement models used for suppression of scene details in the images. Here, x- axis represents the pre-enhanced noise residual values in the wavelet domain and the y-axis represents the post-enhanced noise residual values.

a single noise residual is more likely to be influenced by the scene compared to the reference pattern which is constructed by averaging multiple images. Since iris images exhibit a rich texture, the enhanced SPN scheme can suppress iris-specific structures from the noise residuals.

Phase SPN: The SPN can be modeled as a white Gaussian noise, and so Kang *et al.* [99] hypothesized that whitening in the frequency domain can further attenuate the scene details. The noise residual, W_i, is first whitened in the frequency domain, followed by extraction of the phase component using discrete Fourier transform (DFT), i.e., $\mathbf{W}_i = DFT(W_i)$ and $\mathbf{W}_{\phi i} = \frac{\mathbf{W}_i}{|\mathbf{W}_i|}$. Here, $|\mathbf{W}_i|$ denotes the Fourier magnitude of $\mathbf{W}_i$. The spatial component of $\mathbf{W}_{\phi i}$ is recovered using the inverse DFT (IDFT). The camera reference pattern is finally constructed by averaging the real part of the recovered spatial noise residual as, $\mathbf{K} = real\left(IDFT\left(\frac{\sum_{i=1}^{N}\mathbf{W}_{\phi i}}{N}\right)\right)$.

So far, we have discussed different sensor identification schemes. How can we gauge the robustness of these methods? Can we deliberately confound the sensor identification methods? We explore these questions next.

1.3.1.2 Sensor De-identification Methods

The counter-forensics literature describes techniques that can be used to suppress or perturb the PRNU pattern embedded in an image. This is often referred to as *source anonymization* [64], *i.e.,* obscuring the 'fingerprint' of the source sensor in an image so as to anonymize the origin of the image. Source anonymization can be used as a privacy preservation scheme that is particularly relevant when the sensor-specific details can be used to associate a sensor with its owner. Assuming that each device is typically associated with a single user, device identification can be indirectly used to reveal the identity of the person possessing that specific device [125]. There have been primarily two approaches to perturb the PRNU pattern for this purpose: (i) compression and filtering based schemes, which typically use strong filtering schemes such as, flat-field subtraction [157] or Wiener filtering [28] that can degrade the PRNU pattern leading to incorrect source attribution; and (ii) geometric perturbation based schemes such as 'seam carving' [28, 63] that distorts the alignment between the sensor reference pattern and the test noise residual, thereby impeding the process of correlating the reference pattern with the test noise residual.

In contrast to source anonymization, *PRNU spoofing* not only suppresses the fingerprint of the source sensor, but it also inserts the fingerprint of the target sensor. An adversary may tamper

with the digital evidence to maliciously exculpate a guilty person or worse, incriminate an innocent person. In recent literature, PRNU spoofing has been performed by two methods: (i) PRNU injection and (ii) PRNU substitution. The first method adds the weighted reference pattern of a pre-selected target sensor to the input image, $\boldsymbol{I}$ [78]. The modified image becomes $\boldsymbol{I'} = [\boldsymbol{I} + \boldsymbol{I} \times \gamma \hat{\boldsymbol{K}}_{\boldsymbol{T}}]$. Here, $\hat{\boldsymbol{K}}_{\boldsymbol{T}}$ is the reference pattern of the target sensor T and γ is a scalar parameter. The second method subtracts the PRNU pattern of the source sensor in an image and then adds the PRNU pattern of a target sensor [107]. The modified image is represented as $\boldsymbol{I'} = \boldsymbol{I} - \gamma \hat{\boldsymbol{K}}_{\boldsymbol{S}} + \beta \hat{\boldsymbol{K}}_{\boldsymbol{T}}$. $\boldsymbol{I}$ belongs to the source sensor S, whose reference pattern is $\hat{\boldsymbol{K}}_{\boldsymbol{S}}$. γ and β are scalar terms.

In [154], the authors examine the viability of PRNU spoofing via injection in the context of *iris sensors* operating in the NIR spectrum [77]. In their work, they computed the forged image as $\boldsymbol{I'} = [F(\boldsymbol{I}) + \gamma \hat{\boldsymbol{K}}_{\boldsymbol{T}}]$. Here, $F(\cdot)$ is the wavelet based denoising filter, and γ is a scalar parameter. The authors further performed the triangle test to detect the spoof attack, but did not analyze the impact of the PRNU spoofing on iris recognition performance. In the current literature, adversarial networks have been used for perturbing images with great success [122]. However, a significant bottleneck of deep-learning based techniques is the need for large amount of training data for driving the perturbation process.

1.3.1.3 Joint Biometric-sensor Representation Methods

Biometric recognition systems comprise a *feature extraction* module that elicits a salient feature representation from the acquired biometric data, and a *comparator* module that compares two sets of feature representations to compute a match score [91]. On the other hand, sensor recognition systems extract sensor pattern noise [112] from a set of training images obtained from different sensors to generate *sensor reference patterns*. To deduce the sensor identity of an unknown test image, first its sensor pattern noise is extracted, and then it is *correlated* with the reference patterns. The test image is assigned to the device whose reference pattern yields the highest correlation value.

In [72], the authors used partial face images acquired using smartphones and employed a weighted sum fusion rule at the score level to combine sensor and biometric recognition. Later,

they extended their work to include feature level fusion in [71] and concluded that score level fusion performed comparatively better. In [73], the authors performed HOG-based face recognition and combined it with Photo Response Non-Uniformity-based sensor recognition at the score level. In [15], the authors combined fingerprint recognition with device recognition by performing feature level fusion of minutiae-cylinder-codes with SRAM start-up values. Fusion at the score or feature level is often dependent on the specific biometric modality and the device sensor used. A specific fusion rule producing the best results on a particular biometric and sensor modality (*e.g,* iris and near-infrared sensors) may not yield optimal results on a different modality (*e.g,* face and RGB sensors), and therefore, needs to be tuned separately for each pair of biometric and sensor modalities. Furthermore, feature-level fusion retains the individual biometric and sensor-specific components that can be recovered from the fused representation using appropriate measures. Obtaining the biometric template may compromise the privacy aspect of biometrics. In contrast, the proposed joint representation non-trivially unifies the biometric and sensor-specific features. As a result, typical countermeasures will be ineffective in disentangling the biometric component from the joint representation. This will implicitly preserve the privacy of the biometric component.

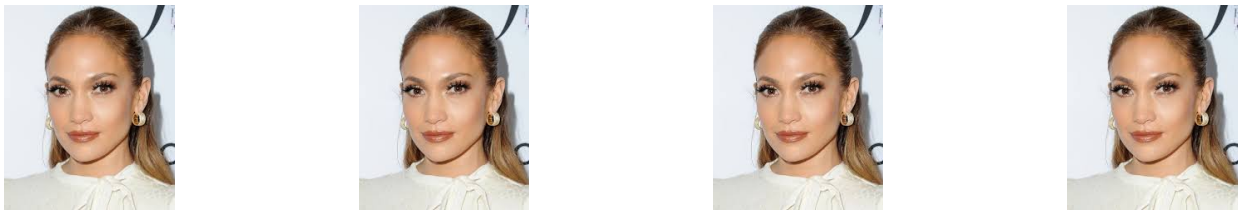
To summarize, we described methods to best estimate sensor-specific traces present in an image and use them to perform sensor identification. We also discussed about PRNU suppression that has applications in the context of privacy preservation. Furthermore, we described the importance of designing an approach to create a joint biometric-sensor representation that can be used to perform biometric and device recognition simultaneously. This has applications int he context of smartphone-based authentication.

Next, we focus on the content-specific analysis of the biometric images that can discriminate between original and modified biometric images.

1.3.2 Content-based forensics

In many applications, the face image of an individual may be subjected to photometric transformations such as brightness adjustment, histogram equalization, gamma correction, etc. These

photometric transformations may be applied in a sequential fashion, resulting in an array of near-duplicate face images (see Figure 1.5). While some of these transformations can be used to improve face recognition [151], others may be maliciously used for image 'tampering' [13]. The availability of inexpensive photo editing tools has resulted in the posting of a large number of near-duplicates on the internet. Identification of the original image from a set of such near-duplicates is important in the context of digital image forensics [65]. Furthermore, inferring the *order of evolution* between a set of near-duplicate images is a challenging but interesting problem [139]. The order of evolution can be represented as an Image Phylogeny Tree (IPT), that indicates the relationship between the root node (original image) and the child nodes (transformed images) via directed links as illustrated in Figure 1.6. The task of image phylogeny is highly challenging as the scope of transformations and the widespread distribution of edited content on online platforms keep evolving. Deducing the IPT from the set of near-duplicates in an automated fashion has the following advantages:



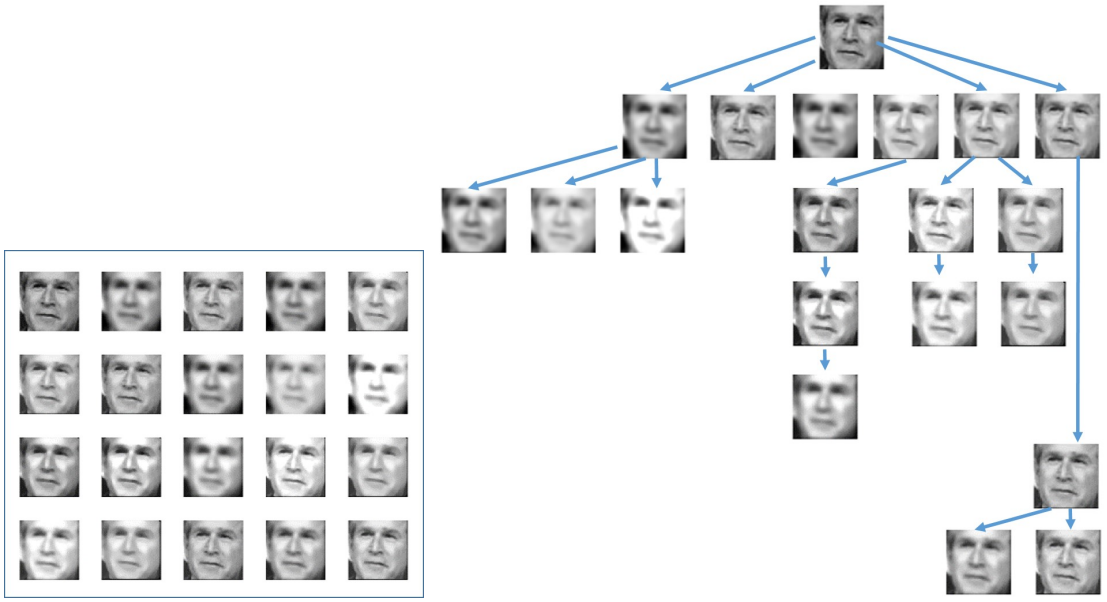
(a) (b) (c) (d)

Figure 1.5: Examples of variations of the same image uploaded on multiple websites with subtle modifications making them appear almost identical.

1. Indication of image tampering: An image can be tampered for a number of reasons. It can be used to airbrush celebrity faces on magazine covers,[2] or to depict fake situations to garner political attention. In either case, the tampered images convey false information. An IPT has directed links, and therefore, can be used to trace an image back to its origin, *i.e.,* the root node that denotes the original image.

2. Preserving chain of custody: Face images can be produced as culpable biometric evidence in legal proceedings [149]. The admissibility of such evidence is contingent on its integrity, and

[2]https://www.cbsnews.com/news/uk-curb-airbrushed-images-keep-bodies-real/



(a) (b)

Figure 1.6: Examples of photometrically related near-duplicate face images. (a) A set of 20 related images and (b) their corresponding Image Phylogeny Tree (IPT). In our work (a) is the input and (b) is the output.

should not have been tampered with. The chain of custody [45] of the digital evidence can be established via (i) hardware identification (the device used to acquire the biometric sample) [24], and (ii) analysis at the image level. IPT construction involves content-based analysis and can be leveraged to determine the authenticity between a pair of biometric samples *i.e.,* the original versus the tampered.

We have primarily motivated the reason for image phylogeny from the perspective of chain of custody preservation of biometric images as culpable evidence [45, 46]. However, our method is not restricted to biometric images only, and can be suitably applied to generic images. Therefore, we can use our method in handling near-duplicates that appear on defaced websites. Website defacement constitutes a hacker typically substituting the original content on the website with images or/and text. The perpetrators often tend to re-use these images, with some alterations in case of mass defacement attacks (see Figure 1.7). Some images bear a logo or an emblem that indicates the affiliation of the hacker to a specific attack group. In such cases, image phylogeny can analyze the evolution of the reused near-duplicates to potentially link it back to the hacker. This

will help in assessing cyberattack patterns and boost cybersecurity.

Figure 1.7: Near-duplicates appearing on defaced websites. Images curated by Zone-H, not meant for public distribution.

1.3.2.1 Image Phylogeny (Tree and Forest) Construction Methods

The task of near-duplicate detection and retrieval (NDDR) is well-studied in the literature [68, 101], and is closely related to the task of image phylogeny. Near-duplicates are defined as semantically similar images but may have slight modifications or may contain portions from multiple source images. In our case, we consider near-duplicates as images belonging to the same individual (also includes images with slightly different pose and expressions in the case of face images) but may have undergone some photometric or geometric transformation to make the images 'appear' almost identical but not exactly identical. We have also quantified the notion of near-duplicity using Structural Similarity Index Measure (SSIM) which computes the similarity between an image pair in terms of luminance, contrast and structural details [163]. The near-duplicates used in our work are reported to have values greater than 75% in terms of SSIM. In cases, where the query image is a composite of multiple donor images, *provenance analysis* [124] has been used, which first identifies

the relevant donor images for a single composite image using a provenance filtering step, and then follow it with a provenance graph construction step for determining the order of modifications. In such cases, undirected phylogeny trees [31] will suffice, since deducing the links is more critical than determining the direction of the link. On the other hand, image phylogeny trees (IPTs) deal with the task of determining directed edges, that is explicitly determining the parent and child nodes such that the edge is directed from the parent node (original image) toward the child node (transformed image). Typically, IPT is considered as a minimal spanning tree (MST), while we interpret IPT as a directed acyclic graph (DAG) by including the ancestral edges, which indicate immediate parents as well as higher level predecessors. Conventionally, an IPT considers a single root node [30,57,58,61,121,133]. Other works have considered the presence of multiple root nodes resulting in Image Phylogeny Forests [62, 128, 129]. A majority of the literature focuses on simple geometric transformations, (cropping, scaling, rotation) and pixel intensity-based transformations (brightness and contrast adjustment, gamma transformation) and compression operations.

Conventionally, IPT construction involves two steps [59, 61]:

(i) Computing an asymmetric (dis)similarity measure between every pair of images in the set.

(ii) Using a tree spanning algorithm to infer the existence of links between image pairs and identify the parent (original) and child (transformed) nodes based on the asymmetric measure.

The first step computes an asymmetric measure which models the relationship between each pair of near-duplicate images in the set. The first step of asymmetric measure computation usually involves geometric registration, followed by color channel normalization and compression matching [61]. Other works focus on using wavelet-based denoising technique [119], and a combination of gradient estimation and mutual information techniques [47] to derive an improved asymmetric measure. Typically, a majority of these methods [61, 119, 124] perform pairwise modeling to compute the asymmetric measure. The objective is to minimize some distance or error function between the pair of images. However, while performing the pairwise modeling, the method *does not* consider the global relationships that the pair of images (under consideration) may share with the remaining images from the set. This is particularly important for an accurate IPT construction

in the second step, which utilizes the asymmetric measure to span the IPT.

Recently, research has steered in the direction of exploring the 'global' relationships by utilizing all the images in the set simultaneously instead of conducting a pairwise analysis. In [38], the authors use a denoising autoencoder to improve the dissimilarity matrix to obtain an accurate IPT. The authors in [32] used a deep neural network to rank the order in which the near-duplicates have been generated by learning the transformation-specific embedding.

Typically, an IPT consists of edges or links which are directed from the original image or the *parent* node towards the transformed image or the *child* node. However, consider a situation where multiple images of the same individual in the same scene are available. Such situations are relatively common. For example, images may have been acquired using different cameras or each image may capture a different facial expression. In this case, transforming each image repeatedly, but independently of the others, will result in multiple IPTs. Each original image will then serve as an individual root, and each root will span a distinct IPT. A collection of such IPTs will constitute an Image Phylogeny Forest (IPF).

In the context of image phylogeny forests, there are basically two types of approaches. (i) Consider IPF construction as an extension of the IPT construction process. Initially, each node is considered as an individual IPT, and then they are successively merged until a terminating criterion is met. The final output is an IPF with multiple IPTs [50, 51, 62]. (ii) Consider IPF construction as a two-step process, where the first step clusters the images, for example, using spectral clustering (each cluster represents an IPT), and the second step constructs the IPT corresponding to each cluster [128]. We focus on the second type of IPF construction in this work.

We would like to point out two observations from the overview of the related literature: (i) In the context of IPT construction, existing work tackle IPT construction by either performing a pairwise node analysis or a global analysis including all the nodes. Alternatively, a graph-based approach can explore both pairwise relationship (first order proximity) as well as relationships with respect to neighboring nodes (higher order proximity). (ii) Limited work has been done in the context of IPF construction, and the state-of the art method uses spectral clustering which fails in multi-scale

examples [29]. In real-world applications, we have no prior knowledge about the number of IPTs, or their scales (number of nodes).

In the following section, we list the thesis contributions from the twin perspective of sensor-based and content-based forensic analysis in the context of biometric images.

1.4 Thesis Contributions

In this thesis, we propose a comprehensive analysis of image forensic schemes in the context of biometric images. To accomplish this objective we have approached it from two different perspectives. Firstly, from the context of sensor-based forensics and secondly, from the context of content-based analysis. Furthermore, we propose work that coalesces our sensor-forensic and content-forensic analyses using a graph-based approach to determine the sequence of evolution of digitally modified biometric images.

1. To understand the sensor-based forensic aspects of biometric images, we studied the feasibility of existing sensor identification schemes in the context of near-infrared ocular images. We focused on Photo Response Non-Uniformity based sensor identification scheme and observed that it can be used to reliably identify iris sensors, with the exception of images that are highly saturated or over-exposed. We demonstrated the impact of photometric transformations (illumination normalization schemes), and observed that some transformations, such as Difference-of-Gaussians (DoG) filtering, can degrade the reliability of PRNU-based sensor identification.

2. To further test the robustness of PRNU-based sensor identification method, we developed two sensor de-identification schemes. They are designed to suppress the sensor-specific traces present in a biometric image while retaining its matching utility. The first method uses an iterative image perturbation routine to perform sensor spoofing. Sensor spoofing involves deliberately confounding the sensor classifier to assign an image to a specific target sensor, different from the original sensor used to acquire it. The second method uses discrete cosine transform to suppress sensor-specific traces and can be used to confound three PRNU-based

sensor classifiers to perform both sensor spoofing and sensor anonymization, where the image is assigned to any random sensor, not necessarily a specific target sensor. Both these methods successfully perform sensor de-identification on images acquired using near-infrared iris and RGB smartphone sensors.

3. To explore the prospect of developing a joint biometric-sensor representation, we developed a method that combines both biometric-specific and sensor-specific traces from a single biometric image. The joint representation can be utilized to simultaneously perform biometric and sensor recognition, as required in smartphone authentication using biometric signatures. We used a deep learning embedding network to capture both biometric and sensor details present in a single biometric image in a one-shot fashion. The joint representation outperformed commercial biometric matchers and PRNU-based sensor recognition in the task of joint identification across three biometric modalities involving multi-spectral sensors. The joint representation was also able to achieve superior performance in the task of joint verification.

4. To address the issue of image phylogeny for near-duplicate biometric images, we examined content-based forensic approaches. To accomplish this objective, we developed two methods. The first method involved a deterministic approach of modeling the photometric transformations between a pair of near-duplicates to discriminate between forward and reverse directions of transformations. By repeating this process for all pairs of near-duplicates in the set, we were able to construct the image phylogeny tree for ocular images subjected to a small set of transformations. The second method involved probabilistic framework by integrating the use of basis functions and likelihood ratio of the estimated parameters obtained form modeling photometric and geometric transformations. We evaluated the method on arbitrary transformations and achieved reliably well performances.

5. To alleviate pairwise modeling required in traditional image phylogeny construction techniques, we used a deep learning-based approach, that combined graph convolutional network

with sensor pattern noise for constructing image phylogeny tree. The graph-based approach provided global analysis by incorporating first order proximity and second order proximity (neighborhood information). Th sensor pattern noise (PRNU) provided local analysis. By leveraging both degrees of analysis, we constructed image phylogeny trees with higher accuracies compared to sate-of-the-art baselines, that demonstrated promising results both on face images as well as images containing natural scenes. We further extended our approach by constructing image phylogeny forests that consisted of multiple image phylogeny trees. We proposed a locally-scaled spectral clustering to identify the number of IPTs and then used graph convolutional network along with PRNU for constructing the forest.

1.5 Thesis Organization

The remaining document is organized as follows:

Chapter 2 introduces sensor forensics in the context of biometrics. We begin with the study of existing sensor identification schemes in the context of ocular images. We further study the impact of photometric transformations on the performance of the sensor identification schemes. This chapter covers the first contribution.

Chapter 3 entails sensor de-identification schemes in the context of privacy preservation. We explore different biometric modalities such as iris and periocular images and analyze how the sensor de-identification schemes impact the biometric recognition and sensor recognition performances. This chapter covers the second contribution.

Chapter 4 describes the joint biometric-sensor representation developed for smartphone sensors. We use a deep learning-based approach to simultaneously learn biometric and sensor-specific traces in a one-shot approach from a single biometric image. This chapter covers the third contribution.

Chapter 5 introduces the content-specific analysis for the task of image phylogeny for digitally modified biometric images. We evaluate the method on iris images. This chapter covers the first (deterministic) method listed in the fourth contribution.

Chapter 6 delves deeper into the task of image phylogeny by combining basis functions and likeli-

hood ratio-based method for constructing image phylogeny trees. We evaluate the proposed method on a suite of image editing operations such as Photoshop and deep learning-based manipulations to test the efficacy of our approach. This chapter covers the second (probabilistic) method listed in the fourth contribution.

Chapter 7 describes the graph-based approach for constructing image phylogeny tree by unifying graph convolutional network and sensor-specific traces. This work combines both sensor and content details present in images for the task of image phylogeny. We further extend it to image phylogeny forests with promising results. This chapter covers the fifth contribution.

Chapter 8 concludes the thesis and provides some steps towards future work.

CHAPTER 2

SENSOR IDENTIFICATION

Portions of this chapter appeared in the following publications:

S. Banerjee and A.Ross, "From Image to Sensor: Comparative Evaluation of Multiple PRNU Estimation Schemes for Identifying Sensors from NIR Iris Images," 5th International Workshop on Biometrics and Forensics, (Coventry, UK), April 2017.

S. Banerjee and A.Ross, "Impact of Photometric Transformations on PRNU Estimation Schemes: A Case Study Using Near Infrared Ocular Images," 6th IAPR/IEEE International Workshop on Biometrics and Forensics, (Sassari, Italy), June 2018.

2.1 Introduction

In this chapter we present two studies. The first study explores sensor identification schemes in the context of biometric sensors, particularly iris sensors. The second study analyzes the impact of photometric transformations on iris sensor identification schemes.

2.2 Study of existing sensor identification schemes on near-infrared ocular images

We have discussed in the previous chapter that there are a number of sensor identification schemes designed for conventional RGB sensors. But we need to analyze whether such schemes can extend to biometric sensors, such as near-infrared iris sensors. This work differs from the existing literature [98, 154] in the following ways: (a) a larger number of sensors are considered (12 sensors); (b) multiple PRNU estimation methods are compared (4 methods); (c) effect of a photometric transformation is investigated; and (d) dataset-specific artifacts are discovered.

Table 2.1: Dataset and sensor specifications.

Name of Dataset	Name of Sensor	Abbreviation	Image Resolution
BioCOP2009 Set I	Aoptix Insight	Aop	640x480
CASIAv3 Interval	Proprietary - not known	Int	320x280
CASIAv3 Lamp	OKI IrisPass-h	OKI	640x480
CASIAv4 Thousand	IrisKing IKEMB100	IK	640x480
CASIAv2 Device2	IrisCam V2	IC	640x480
IITD	JIRIS JPC1000	JPC	320x240
BioCOP2009 Set II	LG iCAM 4000	LG i40	640x480
BioCOP2009 Set III	Crossmatch I SCAN2	ISCAN	480x480
ND_Cosmetic_Contact_Lens_2013	IrisGuard AD100	AD	640x480
ND CrossSensor Iris 2013 Set I	LG2200	LG22	640x480
ND CrossSensor Iris 2013 Set II	LG4000	LG40	640x480
WVU Off-Axis	EverFocus Monochrome CCD	Mon	640x480

2.2.1 Experiments and results for the first study

Below we provide a description of the datasets and experimental protocol used in this work. Then we summarize the results obtained.

2.2.2 Datasets used in the first study

In this work, we use 12 iris datasets that contain images corresponding to 12 different sensors. The details pertaining to the sensors and the datasets are summarized in Table 2.1. The number of images used for reference pattern generation (*i.e.,* the training set) is maintained at 55 per sensor, while for testing, the number of images varied from 100 to 1940. Subjects in the training and test sets were mutually exclusive. Only 55 images were used for reference pattern generation because of the limited number of subjects available in some datasets. For example, the CASIAv2 Device2 dataset contains only 60 subjects; therefore, one iris image from each of 55 subjects was assigned to the training set, and images from the remaining 5 subjects were assigned to the test set.

2.2.3 Experimental methodology used in the first study

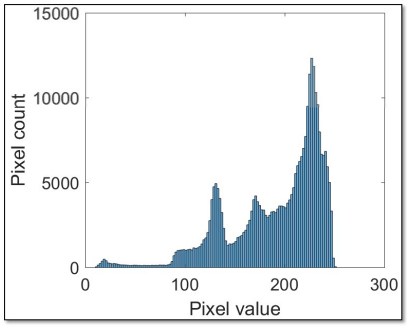
In the first set of experiments, the four PRNU estimation methods were applied on all 12 datasets (details of the implementation are mentioned later in the section). We observed that the performance of all the PRNU estimation methods was poor on BioCOP2009 Set III (Crossmatch) compared to the

other datasets. We investigated the histogram of pixel intensities of images in the individual datasets to discern whether sensor recognition was being unduly impacted by inherent image characteristics. We observed that the histogram of Crossmatch images exhibited a bimodal distribution with a compressed range of pixel values, that is typically associated with contrast adjustment (see Figure 2.1). This prompted us to use the box-cox transformation, which is a series of power transformations, to force the images to conform to an approximate normal distribution. The equation governing the box-cox transformation is as follows:

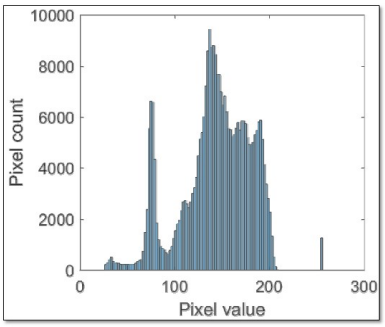
$$output(\lambda) = \begin{cases} \frac{input^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0; \\ log_e(input), & \text{if } \lambda = 0. \end{cases}$$

The value of λ is estimated using maximum log-likelihood function, such that, the distribution of the transformed output closely approximates the normal distribution. In our experiments, λ was predominantly found to be in the interval [-1, 5]. This transformation also aids in studying the effect of photometric processing on PRNU estimation. The individual accuracies of the sensors, as well as the overall accuracies obtained with and without box-cox transformation, are presented in Table 2.2 (for the sake of brevity we have reported the results for Basic SPN alone). The third column in Table 2.2 reports the results obtained after applying the box-cox transformation to the images in the datasets. The last column presents the results after applying the box-cox transformation *only* to images captured using the Crossmatch sensor. The latter was done for investigative purposes only, and not for evaluation purposes.

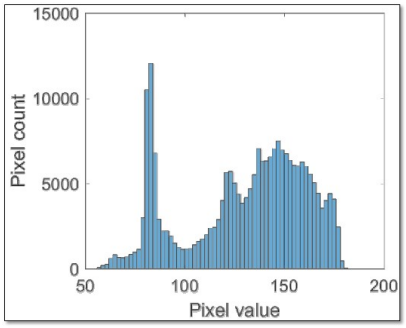
The second set of experiments omitted two datasets - the BioCOP2009 Set III and ND CrossSensor Iris 2013 Set II (the reason for removing the two datasets will be discussed later). In our implementation of Enhanced SPN, the Basic SPN is first extracted and normalized using L_2-norm as described in [52] followed by the application of the enhancement model to the wavelet coefficients. Enhancement Model III (see Figure 4(c)) was used in our work as it was observed to be more robust



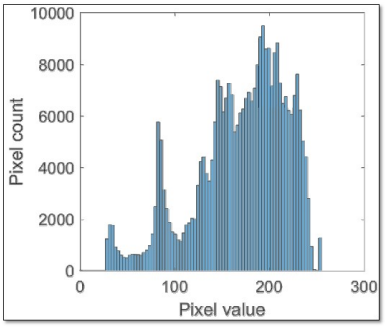
(a) Aop



(b) LG i40



(c) ISCAN



(d) LG40

Figure 2.1: Average pixel-intensity histograms of four sensors. The pixel intensities vary across different sensors indicating diverse image characteristics.

to the variations in α [106]. The enhancement equations are as follows:

$$n_e(i,j) = \begin{cases} 1 - e^{-n(i,j)}, & \text{if } 0 \leq n(i,j) \leq \alpha; \\ (1 - e^{-\alpha})(e^{\alpha - n(i,j)}), & \text{if } n(i,j) > \alpha; \\ -1 + e^{n(i,j)}, & \text{if } -\alpha \leq n(i,j); \\ & \& \quad n(i,j) < 0; \\ (-1 + e^{-\alpha})(e^{\alpha + n(i,j)}), & \text{if } n(i,j) < -\alpha. \end{cases}$$

Here, $n(i,j)$ and $n_e(i,j)$ indicate the original and enhanced noise residual values, respectively (i and j are the indices of the noise residual in the wavelet domain). $\alpha = 6$ in our work. While we did not use L_2 normalization in the first set of experiments, we employed the same enhancement model and same α value in both sets of experiments. The reference pattern is identical to the one used for Basic SPN scheme. For Phase SPN, we did not perform whitening in the frequency domain. This

Table 2.2: Sensor identification accuracies *before* and *after* applying box-cox transformation.

Name of Sensor (Abbreviation)	**Before box-cox (Basic SPN)**	**After box-cox (Basic SPN)**	**After box-cox on Crossmatch images *only* (Basic SPN)**
Aop	100%	91.06%	100%
Int	75.29%	68.24%	74.71%
OKI	100%	91.22%	100%
IK	99.38%	99.64%	99.38%
IC	100%	100%	100%
JPC	100%	99.11%	100%
LG i40	60.83%	80.08%	60.92%
ISCAN	23.16%	49.80%	53.07%
AD	76.17%	94.13%	76.41%
LG22	97.59%	70.19%	97.59%
LG40	88.64%	70.08%	88.26%
Mon	97.78%	82.50%	97.78%
Overall Accuracy	87.26%	88.19%	88.91%

is because all the images used in our experiments are iris images, and spurious frequency responses arising due to variations in scene details is not a critical issue.

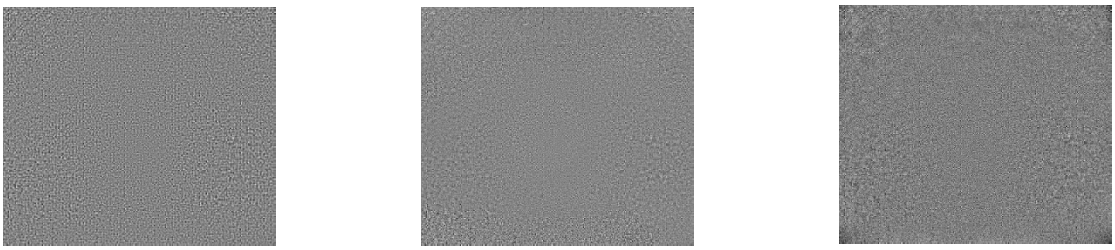
In both experiments, SPN was extracted without subjecting the images (training and test) to any geometric transformation (cropping, resizing, etc.). However, we resized the test noise residuals to the size of the reference pattern for normalized cross-correlation (NCC) computation.

2.2.4 Results and discussion from the first study

Analysis of results on the first set of experiments: The results reported in Table 2.2 show two notable characteristics. Firstly, the accuracy on BioCOP2009 Set III (Crossmatch I SCAN2) improves by 26% after applying box-cox transformation, while it decreases on most of the other datasets. Secondly, BioCOP2009 Set II (LG iCAM4000) witnessed an improvement in performance, while the accuracies on the ND CrossSensor 2013 Sets I and II (LG 2200 and LG 4000) observed a decrease in performance. Therefore, the box-cox transformation improves the performance on the LG iCAM 4000 sensor but at the expense of the other two sensors (LG 2200 and LG 4000). Further investigation indicates that most of the confusion occurred between images from the LG

iCAM 4000 and LG 4000 cameras. The failure of Basic SPN, which is known to be effective in distinguishing between sensors from the same vendor and model, cannot be easily explained in this case. We believe that even though PRNU derived methods try to approximate the SPN via wavelet coefficients, the image histogram also influences the sensor noise in some fashion. The improvement in accuracy of the Crossmatch sensor after the box-cox transformation leads us to believe that the images in BioCOP2009 Set III have perhaps undergone some form of illumination normalization. Moreover, the histogram of images from LG iCAM 4000 and LG4000 reveal that the latter has more saturated pixels making it prone to failure. Due to an incomplete understanding about the extent to which the SPN is impacted by NIR image characteristics and unavailability of pre-processing information, we decided to continue our second set of experiments without the two ambiguous datasets (BioCOP2009 Set III and ND CrossSensor Iris 2013 Set II).

Analysis of results on the second set of experiments: The reference patterns generated using different PRNU estimation schemes are illustrated in Figure 2.2. The effect of enhancing the noise residual is visualized in Figure 2.3. The structure of the eye is evident in the noise residual extracted using Basic SPN which is subdued, but not completely removed, using Enhanced SPN.



(a) Basic SPN (b) MLE SPN (c) Phase SPN

Figure 2.2: Reference patterns for the CASIAv3 Interval dataset estimated using different PRNU estimation schemes. Visual inspection reveals noise like pattern extracted from the training images that are devoid of image content.

The rank-1 confusion matrices obtained for each of the PRNU estimation schemes are reported in Tables 2.3 and 2.4. In Table 2.3, the values to the left of the slash denote the results obtained using Basic SPN and the values to the right indicate the results obtained using MLE SPN. In Table 2.4,

(a) (b)

Figure 2.3: Noise residual from an image captured using the Aoptix Insight sensor. (a) Before enhancement. (b) After enhancement. The application of enhancement model subdues the scene content in the image significantly.

Table 2.3: Rank-1 Confusion Matrix for Basic SPN / MLE SPN based PRNU extraction scheme.

Actual \ Predicted	Aop	Int	OKI	IK	IC	JPC	LGi40	AD	LG22	Mon
Aop	**492/492**	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
Int	7/5	**262/210**	6/16	21/16	13/15	8/39	12/8	1/10	3/9	7/12
OKI	0/0	0/10	**763/727**	0/8	0/6	0/8	0/2	0/0	0/2	0/0
IK	0/0	0/0	1/0	**1928/1940**	0/0	0/0	0/0	11/0	0/0	0/0
IC	0/0	0/0	0/0	0/0	**100/100**	0/0	0/0	0/0	0/0	0/0
JPC	0/0	0/0	0/0	0/0	0/0	**1690/1690**	0/0	0/0	0/0	0/0
LGi40	6/42	40/176	17/48	3/49	4/53	60/174	**878/365**	6/44	14/43	11/45
AD	13/30	33/108	30/24	12/42	11/44	60/153	8/70	**642/301**	12/32	14/31
LG22	0/0	0/0	0/0	0/0	0/0	0/0	0/0	2/0	**527/540**	11/0
Mon	0/0	0/0	0/0	0/0	0/0	0/0	0/0	8/0	0/0	**352/360**

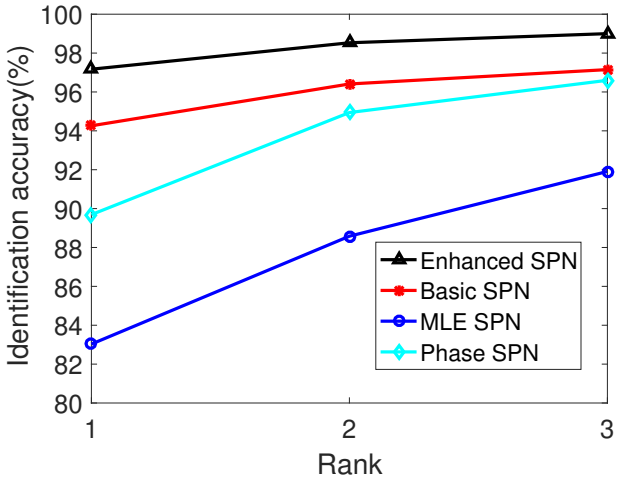
Table 2.4: Rank-1 Confusion Matrix for Enhanced SPN / Phase SPN based PRNU extraction scheme.

Actual \ Predicted	Aop	Int	OKI	IK	IC	JPC	LGi40	AD	LG22	Mon
Aop	**487/492**	0/0	0/0	0/0	0/0	0/0	0/0	5/0	0/0	0/0
Int	7/2	**259/224**	4/11	27/20	14/15	6/40	13/10	2/5	2/7	6/6
OKI	0/0	0/9	**763/743**	0/1	0/2	0/8	0/0	0/0	0/0	0/0
IK	0/0	0/0	0/0	**1934/1940**	0/0	0/0	0/0	6/0	0/0	0/0
IC	0/0	0/0	0/0	0/0	**100/100**	0/0	0/0	0/0	0/0	0/0
JPC	0/0	0/0	1/0	0/0	0/0	**1689/1690**	0/0	0/0	0/0	0/0
LGi40	2/6	9/83	11/11	1/14	2/12	20/117	**945/744**	31/18	12/24	6/10
AD	3/14	2/104	2/19	1/17	0/19	5/140	7/42	**810/431**	4/27	1/22
LG22	0/0	0/0	0/0	0/0	0/0	0/0	0/0	1/0	**531/540**	8/0
Mon	0/0	0/0	0/0	0/0	0/0	0/0	0/0	8/0	0/0	**352/360**

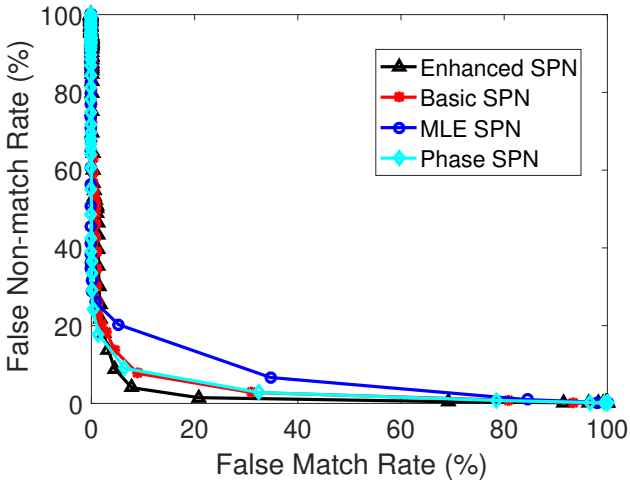
the values before the slash correspond to the Enhanced SPN and values after the slash correspond to the Phase SPN.

Figure 2.4 illustrates the overall performance of each of the PRNU extraction schemes using Cumulative Match Characteristics (CMC) and Receiver Operating Characteristics (ROC) curves. The following observations can be made:

- Enhanced SPN results in the best performance with a rank-1 accuracy of **97.17**%.



(a) CMC



(b) ROC

Figure 2.4: Comparison of overall accuracy of different PRNU extraction schemes using CMC and ROC curves.

• MLE SPN performs the worst (with a rank-1 accuracy of 83.03%). We believe that the zero mean operation in conjunction with Wiener filtering that is used in MLE SPN, over-smoothens the images to the extent that the wavelet coefficients fail to effectively capture the SPN. Iris sensors operating in NIR spectrum do not use CFAs, and therefore, the use of Wiener filtering and zero-mean processing is not necessary in our opinion. The Basic SPN, Enhanced SPN and Phase SPN do not apply these pre-processing operations and exhibit higher accuracies, thereby validating our supposition.

• Phase SPN performs moderately well. Phase SPN performs on par with Basic SPN at Rank-3 (where the correlation score with the correct reference pattern occurs among the top 3 values).

• The poor performance of all the SPN methods on the CASIAv3 Interval dataset may be due to the use of more than one sensor for assembling this dataset.

• The misclassification of images from LG iCAM 4000 (see Tables 2.3 and 2.4) could be due to pixel padding in the former that might have negatively impacted PRNU estimation.

• Application of the box-cox transformation showed a significant increase in performance on the AD100 sensor which had lower performance when using the Basic, Phase and MLE SPN schemes. Upon closer investigation, this may be due to the digital watermark embedded in the images as shown in Figure 2.5. Both the box-cox transformation and Enhanced SPN suppress this artifact thereby leading to improved accuracy.

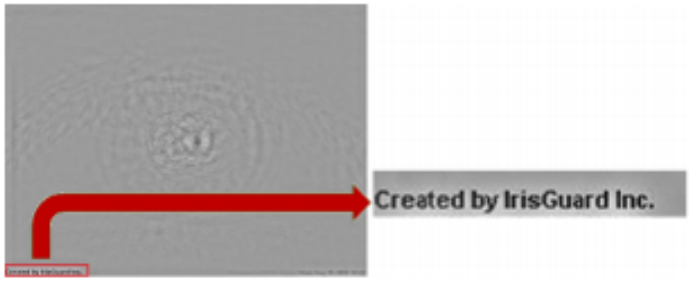


Figure 2.5: Digital watermark present in the reference pattern of AD100 sensor. (The logarithmic transformation has been used here for better visualization).

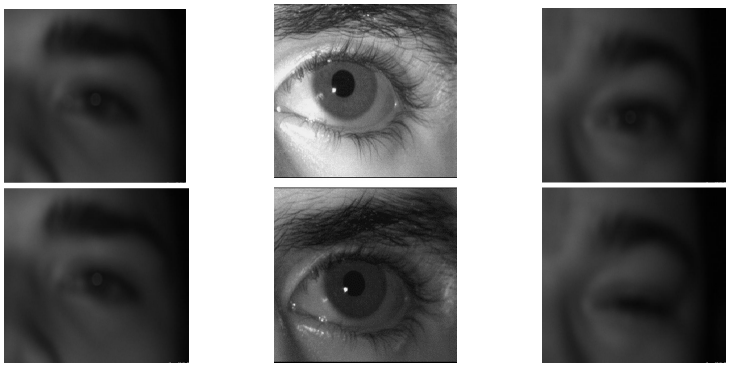
2.2.5 Summary of the first study

The first study investigated the use of PRNU estimation schemes for determining sensor information from NIR iris images. Experiments involving 12 sensors and 4 PRNU estimation schemes showed that device identification can be successfully performed on a majority of the sensors. Modification of the image histograms using box-cox transformation resulted in improved accuracy on some of the sensors but negatively impacted the accuracy of other sensors. Experimental results revealed that Basic SPN and Enhanced SPN performed favorably across most of the sensors and outperformed MLE SPN and Phase SPN by a significant margin. Enhanced SPN performed better than Basic

SPN. We also ascertained that photometric transformations play a role in PRNU-based sensor recognition performance, which will be thoroughly analyzed in the second study described next.

2.3 Analyzing the effect of photometric transformations on sensor identification schemes for ocular images

In the first study, we demonstrated the feasibility of PRNU-based sesnor identification schemes in the context of iris sensors. While sensor forensic schemes have been extensively studied in the context of color images produced by classical digital cameras based on CMOS or CCD technology [41, 70, 76, 112], their applicability to near-infrared (NIR) sensors was only recently established particularly in the context of iris recognition systems [18, 52, 54, 98, 100]. A typical



(a) (b) (c)

Figure 2.6: Examples of Near-Infrared (NIR) ocular images exhibiting (a) defocus blur, (b) uneven illumination and (c) motion blur (due to eyelid movement).[1]

iris recognition system acquires an NIR ocular image, segments the annular iris region from the ocular image, converts this region to a geometrically normalized rectangular entity, and extracts a binary code from the normalized image for matching purposes [48]. Often, the input ocular image is subjected to some illumination normalization schemes in order to address issues such as motion blur, out-of-focus imaging, low resolution and uneven illumination [95]. See Figure 2.6. The goal of such illumination normalization schemes is to improve the recognition accuracy of iris recognition systems by favorably impacting their segmentation and feature extraction modules.

[1] Images are acquired using Panasonic BM-ET100US (b) and Cogent sensors (a & c).

In this study, we examine the following question: Do these commonly applied photometric transformation schemes impede the performance of *iris sensor identification* algorithms? Such a study has the following benefits:

- It would help in better understanding the robustness of different PRNU-based schemes to commonly applied illumination normalization routines in the iris recognition domain. This is particularly important in situations where the original raw image is not available for forensic purposes, but the processed image is available (e.g., when pre-processing is accomplished using hardware).
- In recent literature, the possibility of combining ocular biometric recognition with device (sensor) identification has been proposed for enhanced security [71], by using the *same* ocular image for both device identification and ocular recognition. Since the photometric normalization schemes considered in this work are known to positively impact *biometric* recognition, it behooves us to determine the nature of their impact on *device* identification.

In this work, we evaluate the effect of photometric transformation on multiple PRNU-based sensor identification techniques, and use Jensen-Shannon based divergence measure to explain the rationale behind the variation in sensor identification performance.

In the current work, we advance our understanding of PRNU-based sensor identification schemes by considering multiple photometric transformations and analyzing the effect of such transformations on sensor identification accuracy, in the context of NIR ocular images. Further, we develop an explanatory model to determine a causal relationship between photometric transformations and their impact on the performance of PRNU algorithms.

The principal contributions of this work are as follows: a) investigating the effect of seven illumination normalization schemes (the terms *illumination normalization*, *photometric transformation* and *image enhancement* have been used interchangeably in the paper) on sensor identification performance; b) conducting experiments using 11 sensors and 4 PRNU estimation schemes; and c) using the Jensen-Shannon divergence measure to explain the impact of photometric transformations

on the wavelet denoised pixel intensity distribution (discussed later) and, subsequently, on sensor identification.

2.3.1 Photometric Transformation

Variations in ambient lighting conditions, coupled with unconstrained image acquisition, result in challenging ocular images as depicted in Figure 2.6. Occlusions due to eyelid movement, motion blur, de-focus blur, poor resolution and varying degrees of illumination can significantly impact iris segmentation and iris recognition processes [95]. A large number of illumination normalization schemes have been demonstrated to improve iris and periocular recognition performance [95, 97, 127, 147]. The relevance of the seven ocular image enhancement schemes considered in our work is discussed next.

Homomorphic Filtering: Homomorphic filtering is most commonly used for removing non-uniform illumination in images by applying a high-pass filter in the frequency domain to images subjected to a logarithmic transformation [79]. Issues arising due to uneven illumination, as depicted in Figure 2.6(b), can be addressed by applying a high pass Butterworth filter after the logarithm transformed image is converted to the frequency domain using Fourier transform. Singh *et al.* [147] used homomorphic filtering to improve the performance of iris recognition on the NHCI database.

Gamma correction: Gamma adjustment is typically used to increase the contrast of images acquired in low illumination conditions [150]. This photometric transformation produces the output image as a power, denoted by a parameter γ, of the input image pixel values. Jillela *et al.* [95] employed gamma correction for improving the contrast of images in the FOCS database for periocular recognition. The range of γ studied in our work is [0.1, 2.1].

Contrast Limited Adaptive Histogram Equalization (CLAHE): Histogram equalization has been shown to aid periocular recognition [94]. CLAHE tessellates the image into patches and performs adaptive histogram equalization on each of these patches by clipping the pixel intensity

[2]Original image was acquired using CASIA-IrisCam V2 sensor [6].

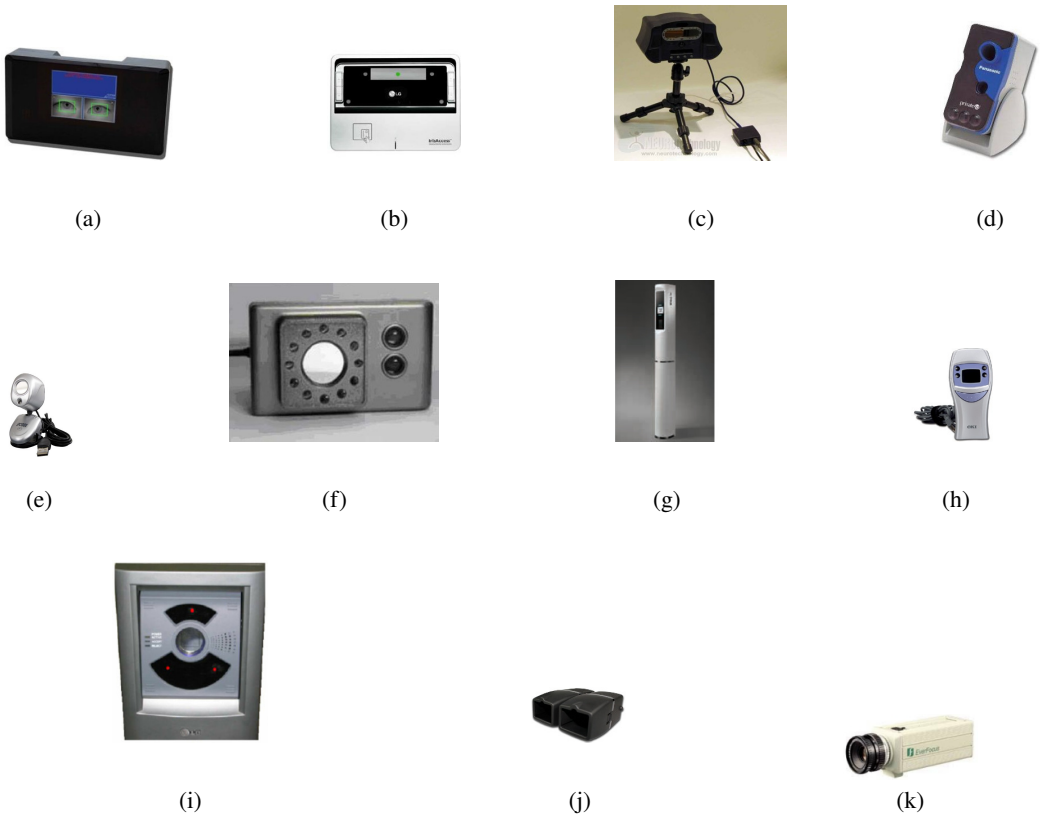


Figure 2.7: Examples of iris sensors considered in this work. (a) IrisKing IKEMB100, (b) LG 4000, (c) IrisGuard-IG-AD100, (d) Panansonic-BM-ET100US Authenticam, (e) JIRIS JPC1000, (f) CASIA IrisCam-V2, (g) Aoptix Insight, (h) OKI IrisPass-h, (i) LG 2200, (j) Cogent and (k) Everfocus Monochrome CCD.

values exceeding the user defined contrast limit [164]. Finally, it aggregates the patches using bilinear interpolation. In our experiments, the size of a patch is 8×8 and the contrast limit is set to 0.01.

Discrete Cosine Transform (DCT): Illumination invariance can be achieved in the logarithmic DCT domain by discarding low frequency DCT coefficients, which captures the illumination of the image [43]. This process operates like a high pass filter. Juefei-Xu and Savvides applied this illumination normalization for robust periocular recognition on NIST's FRGC version 2 database [97].

Difference of Gaussians (DoG): DoG filter closely approximates the Laplacian of Gaussian (LoG) filter in a computationally efficient manner [89]. DoG uses Gaussian filters having different scales or filter sizes. The difference between the two filtered outputs, corresponding to Gaussian

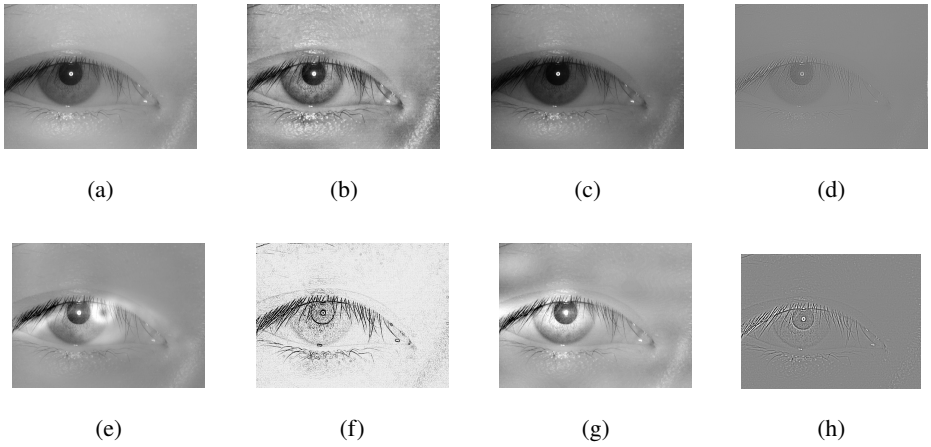


Figure 2.8: An example of an NIR iris image subjected to seven illumination normalization schemes. (a) Original, (b) CLAHE, (c) Gamma correction, (d) Homomorphic filtering, (e) MSR, (f) SQI, (g) DCT normalization and (h) DoG.[2]

filtering of the image using two different scales, is computed. The difference computed above, which is the final output is devoid of the illumination variations present in the original image. DoG filtering has been used to compensate for illumination variations in the context of periocular recognition on the FRGC version 2 database [97]. The two filter sizes that were used in our work are σ_1 =1 and σ_2 =2, where σ_i denotes the standard deviation.

Multi-Scale Retinex (MSR): Multi-Scale Retinex (MSR) [96] uses smoothing kernels of different sizes, and combines the outputs of Single Scale Retinex (SSR) to remove the halo-like artifacts produced in images transformed using a kernel of a single scale. The retinex algorithm was applied to UBIRIS v2, FRGC and CASIAv4-Distance datasets to improve the quality of ocular images [152]. MSR proved to be the best illumination normalization scheme in [97]. Three scales (standard deviations), viz., σ = [7, 15, 21], were used in our work to retain the fine details present in the scene as well as maintain the visual aesthetics of the image.

Single-Scale Self Quotient Image (SQI): SQI is closely related to MSR. It is based on the Lambertian model and the concept of quotient image [160]. The illumination invariant representation can be obtained as the quotient of the original image and the smoothed version of the original image. The halo-like artifacts produced in MSR is typically due to the use of an isotropic Gaussian smoothing kernel. This problem is resolved in SQI using a weighted anisotropic Gaussian

smoothing kernel [160]. SQI was used for illumination normalization on the UBIPosePr dataset for unconstrained periocular recognition [131].

Figure 2.8 illustrates the effect of the aforementioned photometric normalization schemes on a sample ocular NIR image. As evident from Figure 2.8(e), MSR is not able to remove the halo artifacts completely, and these anomalies persist in Figure 2.8(g), where DCT based normalization scheme is used.

2.3.2 Experiments and results for the second study

In this section, we review the datasets used in our work, followed by a discussion of the results observed in the second study.

2.3.3 Datasets used in the second study

We use 11 iris datasets corresponding to 11 different sensors. The details concerning the sensors and the datasets are outlined in Table 3.1. The sensor reference pattern is generated using 55 training images per sensor and the number of test images varied from 528 to 940 per sensor. The subjects in the training and test sets were mutually exclusive.

Table 2.5: Rank-1 Sensor Identification Accuracies (%). The value enclosed in parentheses indicates the difference in accuracy when compared to that obtained using the original images. Note that in all cases, the reference pattern for each sensor is computed using the unmodified original images.

	PRNU Estimation Schemes			
Photometric Transformation	**Basic SPN**	**Enhanced SPN**	**Phase SPN**	**MLE SPN**
Original	**96.43**	**98.87**	**94.89**	**97.10**
Homomorphic	92.38(-4.05)	93.38(-5.49)	93.37(-1.52)	97.79(+0.69)
CLAHE	95.75(-0.68)	97.78(-1.09)	94.51(-0.38)	96.43(-0.67)
Gamma	96.53(+0.10)	98.03(-0.84)	95.41(+0.52)	97.60(+0.50)
DCT normalization	95.54(-0.89)	97.01(-1.86)	96.20(+1.31)	97.35(+0.25)
DoG	92.81(-3.62)	92.77(-6.10)	90.28(-4.61)	90.42(-6.68)
MSR	96.31(-0.12)	96.18(-2.69)	98.16(+3.27)	98.20(+1.10)
SQI	95.04(-1.39)	96.82(-2.05)	94.47(-0.42)	94.00(-3.10)
Average	**94.90(-1.53)**	**95.99(-2.88)**	**94.63(-0.26)**	**95.97(-1.13)**

Table 2.6: Jensen-Shannon divergence values computed between the wavelet-denoised versions of the original and the photometrically transformed images.

Transformations	**Name of Sensor**											**Mean and**
	Aop	**OKI**	**IC**	**IK**	**JPC**	**Cog**	**Pan**	**AD**	**LG22**	**LG40**	**Mon**	**Std. Deviation**
Ori-CLAHE	0.5543	0.1018	0.2092	0.1053	0.3165	0.0982	0.4027	0.4403	0.4465	0.6798	0.0912	0.3133 ± 0.2071
Ori-DCT	0.5325	0.0724	0.1330	0.0824	0.5179	0.1033	0.3522	0.3402	0.3314	0.3838	0.0571	0.2642 ± 0.1801
Ori-DoG	0.4791	0.0970	0.1370	0.0754	0.3074	0.0760	0.3134	0.2516	0.3263	0.3230	0.0397	0.2205 ± 0.1423
Ori-Gamma	0.4559	0.0836	0.0960	0.0499	0.2310	0.1143	0.2282	0.2399	0.3062	0.2561	0.1036	*0.1968 ± 0.1211*
Ori-Homo	0.6184	0.0947	0.0769	0.1252	0.3584	0.3366	0.3979	0.3894	0.4949	0.5962	0.0612	0.3227 ± 0.2057
Oril-MSR	0.5602	0.1231	0.1657	0.2409	0.7459	0.0910	0.4060	0.3682	0.5440	0.4311	0.0678	0.3404 ± 0.2217
Ori-SQI	0.7320	0.1191	0.2915	0.1880	0.3522	0.1361	0.5478	0.6849	0.5867	0.8173	0.1304	**0.4169 ± 0.2644**

2.3.4 Experimental methodology and results for the second study

Experiments were conducted on 9,626 images (605 for training and 9,021 for testing) and the results are reported in terms of Rank 1 identification accuracy. Rank 1 accuracy corresponds to the proportion of test images assigned to the correct sensor class, *i.e.,* those images that yield the highest NCC when compared against the reference pattern of the sensor they actually originated from. Note that in all experiments, the sensor reference pattern was always computed using the *original* training images and not the photometrically modified images. Table 2.5 reports the sensor identification accuracies for each PRNU method. Inferences drawn from this table are presented below:

- **Observation#1:** The application of photometric transformations marginally decreases the sensor identification performance of the 4 PRNU estimation schemes considered in this work. Note that the photometric schemes considered herein are applicable in the context of iris and periocular recognition.

- **Observation#2:** Enhanced SPN emerges to be the *most robust* to illumination normalization methods among the 4 PRNU estimation schemes, closely followed by MLE SPN. The robustness is assessed by computing the average of the rank-1 sensor identification accuracies corresponding to the 7 photometric transformations. Enhanced SPN resulted in the highest average identification accuracy of 95.99% followed by MLE SPN with an average of 95.97%. Basic SPN yielded 94.90% and Phase SPN resulted in 94.63%.

- **Observation#3:** Photometric transformations were observed to improve the sensor identification accuracy of the Phase SPN method. Multi-Scale Retinex improved the accuracy by 3.27%, DCT normalization boosted the accuracy by 1.31% and Gamma correction marginally improved it by 0.52%.

- **Observation#4:** DoG filtering resulted in degradation of sensor identification accuracy by 3.62% for Basic SPN, by 6.10% for Enhanced SPN, by 4.61% for Phase SPN and by 6.68% for MLE SPN. It was closely followed by SQI which degraded the sensor identification accuracy by 1.39% for Basic SPN, by 2.05% for Enhanced SPN, by 0.42% for Phase SPN and by 3.10% for MLE SPN. Based on the results of this work, it is evident that some illumination normalization schemes which help in improving iris recognition performance, can negatively impact the performance of sensor identification algorithms.

- **Observation#5:** Gamma transformation and MSR have marginal influence on all the PRNU estimation schemes, as seen by the difference-in-performance values enclosed in parentheses.

The results are further visualized from two perspectives. First, CMC curves are presented in Figure 2.9 which depict the effect of *each photometric normalization scheme* on the PRNU estimation techniques. Secondly, ROC curves are presented in Figure 2.10 indicating the degree of robustness of *each PRNU estimation algorithm* when subjected to different illumination normalization methods. These two curves reinforce the observations made above.

Next, we address the following question: *Is there an explanatory model which can describe the performance of the PRNU estimation schemes in the presence of photometrically transformed images*? The next section utilizes a statistical measure to explain the variations in the performance of the sensor identification algorithms when applied to photometrically modified images.

2.3.5 Analysis and explanatory model for the second study

The results in the previous section indicate that PRNU estimation schemes are able to recover sensor information reliably for some commonly used illumination normalization schemes applied to ocular

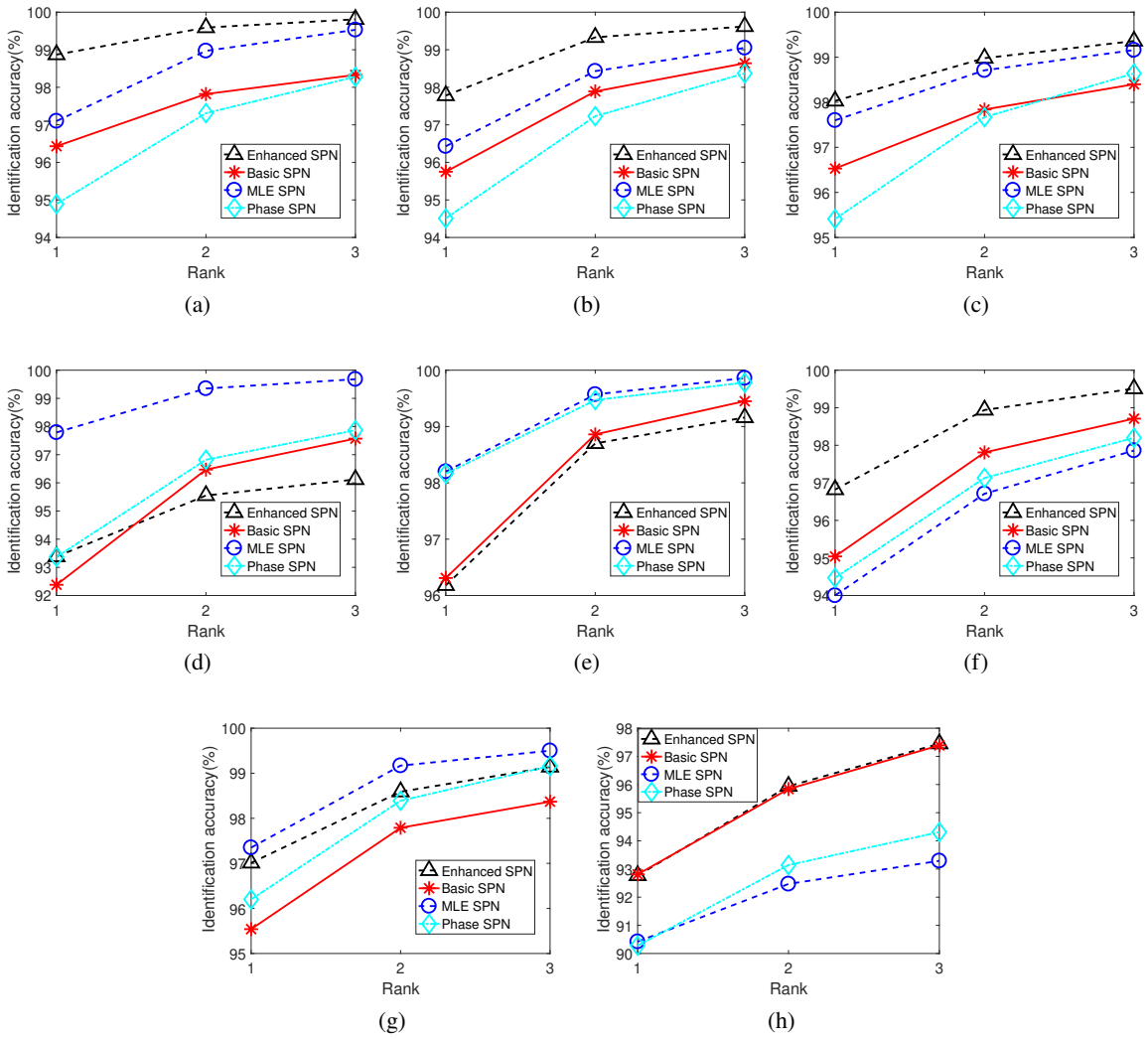


Figure 2.9: Cumulative Matching Characteristics (CMC) curves depicting the effect of different illumination normalization processes on PRNU estimation techniques. (a) Original, (b) CLAHE, (c) Gamma correction, (d) Homomorphic filtering, (e) MSR, (f) SQI, (g) DCT normalization and (h) DoG.

images, barring DoG filtering and SQI transformation. In this section, we study the probability distribution of pixel intensities, *i.e.,* the normalized histograms of the original image and the photometrically transformed images after being subjected to the wavelet based denosing filter, to provide a principled analysis of the performance of PRNU-based sensor identification algorithms. We hypothesize that the degree of disparity between the histograms of the denoised original images and the denoised transformed images will provide insight into the general performance of PRNU estimation algorithms on photometrically modified images. The four sensor identification schemes

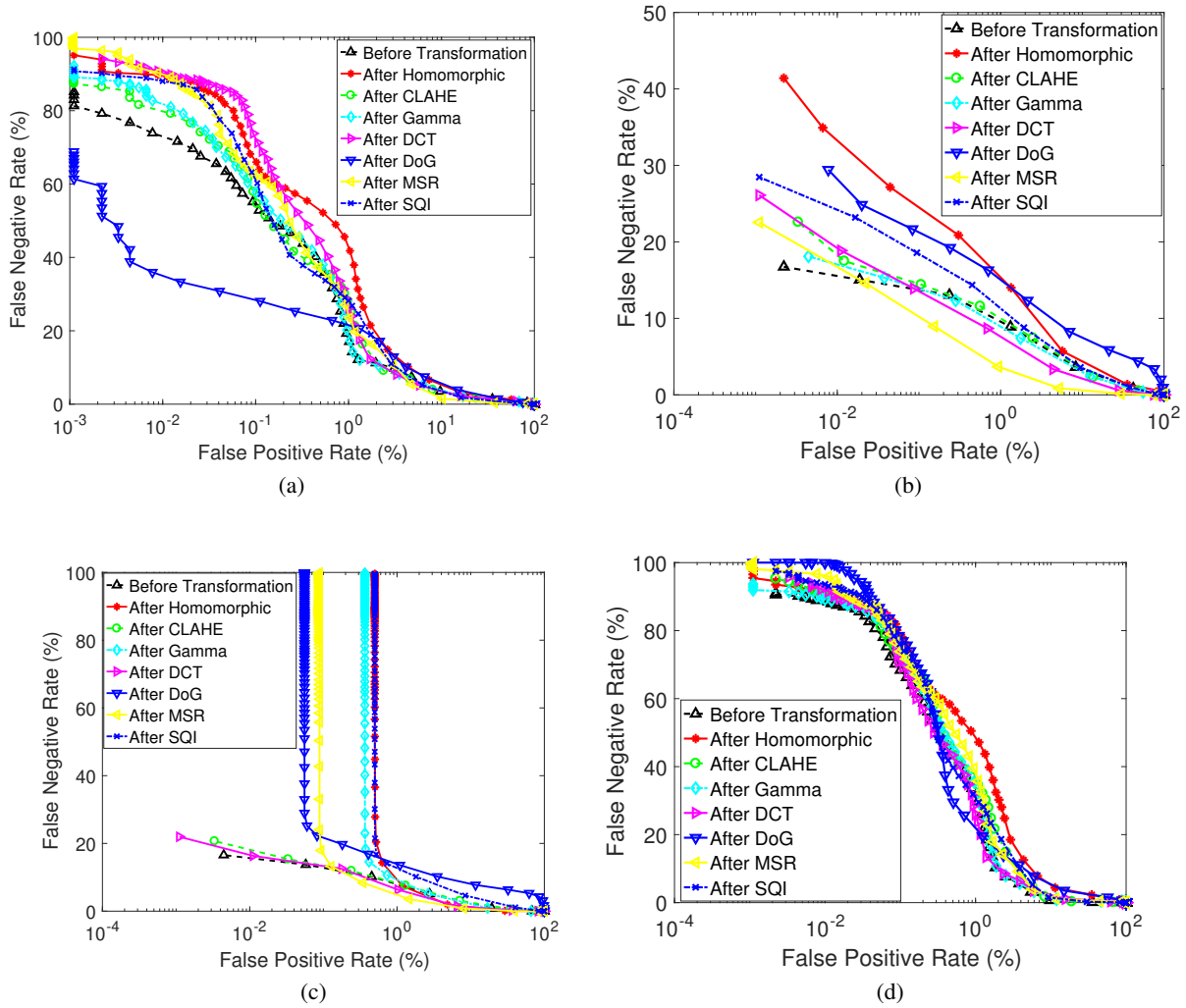


Figure 2.10: ROC curves depicting sensor identification performance of photometrically transformed images. (a) Basic SPN, (b) Phase SPN, (c) MLE SPN and (d) Enhanced SPN.

used in this work are not applied to the raw images directly; rather, the images (both original and transformed) are first subjected to wavelet based denoising, followed by PRNU estimation in the wavelet domain. Thus, it is necessary to consider the denoised images, instead of the raw images, to develop a suitable explanatory model. To this end, we employed the Jensen-Shannon (JS) divergence to compute the dissimilarity between denoised image histograms corresponding to the original image and the photometrically modified image. JS divergence is a symmetric and smoothed version of Kullback-Liebler divergence and yields a finite value [146]. Given two probability distributions, P_o and P_t, the JS divergence is computed as follows, $JS(P_o||P_t) =$

$\mathbf{H}(\frac{1}{2}P_o+\frac{1}{2}P_t)-\{\frac{1}{2}\mathbf{H}(P_o)+\frac{1}{2}\mathbf{H}(P_t)\}$. Here, $\mathbf{H}$ indicates the Shannon entropy measure corresponding to a random variable, say X, and is computed as $\mathbf{H}(X) = -\sum_i p_i \log_2(p_i)$. Here, $p_i = \mathrm{P}[X = x_i]$. The JS measure is bounded between 0 and 1 (0 corresponds to identical distributions and 1 indicates high dissimilarity). Thus, JS divergence computes the average entropy of the two distributions: higher the entropy value, the more dissimilar are the two distributions.

First, we generated the probability distributions (*i.e.,* the normalized histograms) of the denoised original and denoised transformed images.[3] Next, we compute the JS divergence between the two probability distributions. Finally, the JS values corresponding to all the images are averaged to compute a single JS measure value for a given sensor. Table 2.6 reports the JS divergence pertaining to different transformations for each of the 11 sensors. The average and standard deviation of the JS values corresponding to the 11 sensors are computed for each of the photometric transformations. The highest divergence value corresponding to a particular transformation is bolded, while the lowest divergence value is italicized. Some important observations from Table 2.6 are summarized below.

- **O**bservation#1: Gamma transformation resulted in the least JS divergence value. It indicates that the normalized histograms of denoised original and Gamma transformed images are *highly* similar. So it is not surprising that Gamma transformation resulted in only a marginal degradation in sensor identification accuracy as evident from the fourth row in Table 2.5.

- **O**bservation#2: 7 out of 11 sensors reported maximum divergence values for the SQI transformation, which resulted in the second worst degradation in Rank 1 accuracy (note 8^{th} row in Table 2.5), trailing just behind DoG.

- **O**bservation#3: The overall results indicate that pixel intensity distributions have an important role to play with regards to PRNU.

[3]Note that PRNU estimation schemes do not require the input scene to be geometrically aligned, since PRNU is a function of the pixel *location* in the original image.

In summary, Enhanced SPN and MLE SPN are robust to most of the illumination normalization schemes used by periocular or iris matchers. Both these methods use $L2-$normalization of noise residuals to account for the variations arising due to constructional differences of sensors [100], which possibly facilitates a more accurate PRNU estimation. Gamma correction and MSR can be used for ocular image enhancement without impairing the performance of the sensor identification module. SQI and DoG filtering, on the other hand, degrade the performance of sensor identification algorithms.

2.3.6 Summary of the second study

This work investigated the impact of photometric transformations on PRNU estimation schemes and employed an explanatory model to understand their performance in the presence of photometrically modified images. Iris recognition systems typically use illumination normalization to enhance ocular images. In this work, photometric transformations which are known to positively impact ocular recognition have been considered for experimental analysis. Experiments involving 7 ocular enhancement schemes and 4 PRNU estimation schemes indicate that Enhanced SPN and MLE SPN are robust to a majority of the illumination normalization schemes considered in this work, and that DoG filtering and SQI can be detrimental for sensor identification (see Section 2.3.4). The explanatory model indicates that those photometric transformations causing significant deviation of the distribution pertaining to the denoised photometrically modified image from the pixel intensity distribution of the denoised original image can negatively impact sensor identification performance. The relative dissimilarity between the distributions pertaining to the denoised original and photometrically transformed images was quantified using the Jensen-Shannon divergence which explained the performance of sensor identification algorithms in presence of photometric transformations.

2.4 Summary

In this chapter, we focused on Photo Response Non-Uniformity (PRNU)- based sensor identification method. We studied the feasibility of using PRNU for biometric sensor identification, particularly, in the context of near-infrared iris sensors. We observed that the performance of the sensor identification scheme is impacted by the underlying distribution of the pixel intensities. Enhanced PRNU achieved 97.17% Rank 1 iris sensor identification accuracy when evaluated on ten iris sensors. We further studied the influence of photometric transformations, that can alter the pixel intensity distributions of the images on such sensor forensic schemes. We observed that some transformations such as Difference-of-Gaussians filtering degraded sensor identification accuracies.

CHAPTER 3

SENSOR DE-IDENTIFICATION

Portions of this chapter appeared in the following publications:

S. Banerjee, V. Mirjalili and A. Ross, "Spoofing PRNU Patterns of Iris Sensors while Preserving Iris Recognition," 5th IEEE International Conference on Identity, Security and Behavior Analysis, (Hyderabad, India), January 2019.

S. Banerjee and A. Ross, "Smartphone Camera De-identification while Preserving Biometric Utility," 10th IEEE International Conference on Biometrics: Theory, Applications and Systems, (Tampa, USA), September 2019.

3.1 Introduction

In the previous chapter, we analyzed whether sensor identification schemes, particularly PRNU-based schemes can be used to aid in sensor recgnition for biometric images. In this chapter, we would like to test the robustness of the schemes. More importantly, whether these sensor recognition schemes can be deliberately confounded. To this end, we have proposed two strategies in this chapter. The first strategy involves altering the images for sensor de-identification in iris sensors. The seocnd strategy involves sensor de-identification for smartphone sensors in the context of partial face images.

3.2 Sensor de-identification for iris sensors

Given the forensic value of PRNU in determining the origin of an image (*i.e.* the sensor or device that produced it), we explore if it is possible to alter an image such that its source, as assessed by a PRNU estimation scheme, is confounded. We impose two constraints:

1. The modified image must spoof the PRNU pattern of a pre-specified target sensor.

2. The biometric utility of the modified image must be retained, viz., the modified ocular image

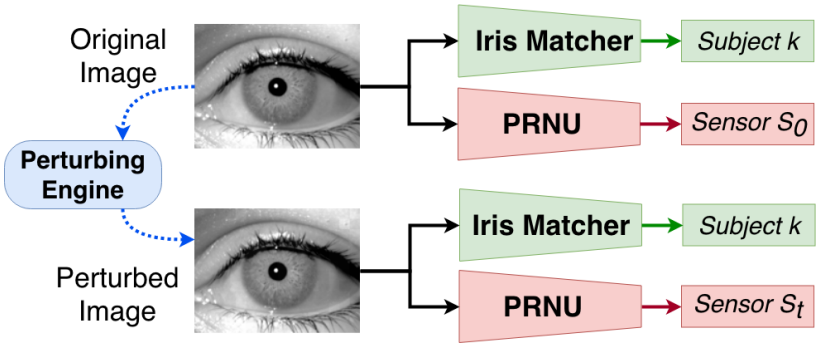


Figure 3.1: Illustration of the objective of the proposed method, *i.e.*, to perturb an ocular (iris) image such that its PRNU pattern is modified to spoof that of another sensor, while not adversely impacting its biometric utility.

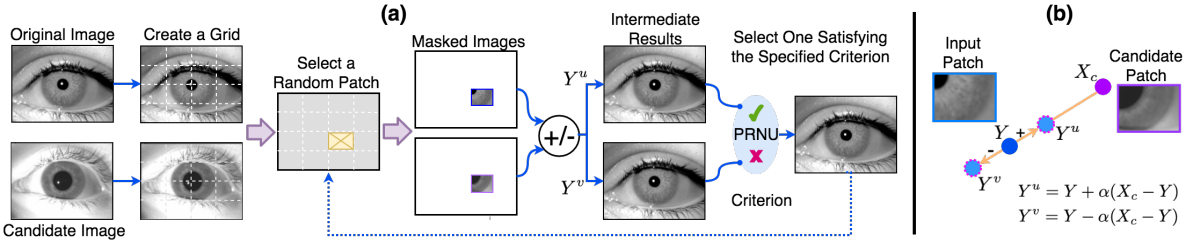


Figure 3.2: The proposed algorithm for deriving perturbations for the input image using the candidate image. (a) Steps involved in modifying the original image from the source sensor using a candidate image from the target sensor (see Algorithm 1), and (b) role of the candidate image in the perturbation engine (see Algorithm 2).

must match successfully with the original image.

This kind of attack can be considered as a 'targeted attack', since the sensor whose PRNU pattern has to be spoofed is pre-specified. In the literature, it is also referred to as fingerprint-copy attack [78, 154], because the objective is to copy the sensor pattern or 'fingerprint' corresponding to the target sensor to an image acquired using a different source sensor. The proposed work has two distinct benefits. Firstly, it allows us to assess the feasibility of PRNU spoofing from a counter-forensic perspective. The widespread use of forensic techniques for examining the validity and origin of digital media [75, 105] necessitates the study of attacks that can potentially undermine the performance of such forensic methods. For example, an adversary may maliciously attempt to link an image to a different camera in an effort to mislead law enforcement investigators [78].

Secondly, establishing the viability of such spoof attacks would promote the development of more robust PRNU estimation schemes [115]. In addition, effective methods to detect such attacks can be developed if the process of spoofing is better understood. Figure 3.1 summarizes the objective of this work.

The remainder of the chapter is organized as follows. We present two works, the first work involves sensor de-identification for iris sensors using iterative image perturbation routine, and the second work involves sensor de-identification in the context of smartphones.

3.2.1 Perturbing the PRNU Pattern for iris sensors

In the first work, our objective is to perform PRNU spoofing in a principled manner, that works for any arbitrary pair of source and target iris sensors. In addition, we wish to retain the biometric utility of the PRNU-spoofed image. The task of spoofing can be potentially accomplished through different techniques, an example will be the use of adversarial networks that have been successfully utilized for perturbing images in the current literature [122]. However, a significant bottleneck of deep-learning based techniques is the need for large amount of training data for driving the perturbation process. We will demonstrate the success of the proposed PRNU spoofing scheme using small number of images (<1000).

3.2.2 Proposed method

In this section, we formally describe the objective and the method used to address this objective.

3.2.2.1 Problem formulation

Let X denote an NIR iris image of width w and height h, and $\boldsymbol{S} = \{S_1, S_2, .., S_n\}$, denote a set of n sensors. Let $\phi(X, S_i)$ be the function that computes the normalized cross-correlation (NCC) between the noise residual of X and the PRNU reference pattern of sensor S_i. Then, the sensor label for the input iris image X can be determined using $\arg\max_i\{\phi(X, S_i)\}$. Furthermore, let M

Table 3.1: Specifications of the datasets used in this work.

Dataset	Sensor Name (Abbreviation)	No. of Images Used (Training set+Testing set)	No. of Subjects
BioCOP 2009 Set I	Aoptix Insight (Aop)	995 (55+940)	100
IITD [1]	Jiristech JPC 1000 (JPC)	995 (55+940)	100
CASIAv2 Device2 [6]	CASIA-IrisCamV2 (IC)	995 (55+940)	50
IIITD Multi-spectral Periocular (NIR subset) [2]	Cogent (Cog)	588 (55+533)	62
ND CrossSensor Iris 2013 Set II [3]	LG 4000 (LG40)	615 (55+560)	99
MMU2 [4]	Panasonic BM-ET 100US Authenticam (Pan)	55 (55+0)	6
ND Cosmetic Contact Lens 2013 [3]	IrisGuard IG AD100 (AD)	55 (55+0)	4
WVU Off-Axis	EverFocus Monochrome CCD (Ever) [144]	55 (55+0)	7
CASIAv2 Device 1 [6]	OKI IrisPass-h (OKI)	55 (55+0)	3
CASIAv4-Iris Thousand subset [7]	IrisKing IKEMB100 (IK)	55 (55+0)	3
ND CrossSensor Iris 2013 Set I [3]	LG 2200 (LG22)	55 (55+0)	5

be a biometric matcher where $M(X_1, X_2)$ determines the match score between two iris samples X_1 and X_2. Given an input iris image X acquired using sensor S_o, a candidate image X_c from the target sensor S_t, and an iris matcher M our goal is to devise a perturbation engine Ψ that can modify the input image as $Y = \Psi(X, X_c)$ such that $\phi(Y, S_o) < \phi(Y, S_t)$, and thereby predict S_t as the sensor label of the perturbed image Y, while the iris matcher, M, will successfully match Y with X. As a result, the target sensor will be spoofed, while the biometric utility of the image will be retained. This implies that the match score between a pair of perturbed images $[M(Y_1, Y_2)]$ as well as that of a perturbed sample with an original sample, $[M(X_1, Y_2)]$ and $[M(Y_1, X_2)]$, are expected to be similar to the match scores between the original samples $[M(X_1, X_2)]$. The steps used to achieve this task are described next.

3.2.2.2 Deriving perturbations and PRNU Spoofing

Given a single image X from the source sensor S_o, a gallery of images $G = \{X_1, ..., X_L\}$ from the target sensor S_t, and a set of K random patch locations $P = \{p_1, ..., p_K\}$, we first select a candidate image, X_c, $c \in [1, \cdots, L]$, from the gallery to perturb the input image. The candidate image is selected from the gallery such that it is maximally correlated with the input image X. To accomplish this goal, we select 10 patches in the input image, each of size 10×10 (*i.e.* $K = 10,\ h_p = 10,\ w_p = 10$ in Algorithm 1). Now, we compute the average pixel intensity in each of these patches and create a K-dimensional vector $\boldsymbol{v}_X$. Next, for each of the L gallery images,

Algorithm 1: Selection of the candidate image

1: **Input:** An image X from sensor S_o, a gallery of images $G = \{X_1, ..., X_L\}$ from the target sensor S_t
2: **Output:** A candidate image, X_c, selected from the gallery
3: Set static parameters $K = 10$ (number of random patches) and $w_p = 10$, $h_p = 10$, (patch width and height).
4: Generate a set of K random patch locations $P = \{p_1, \cdots, p_K\}$, where each patch size is $h_p \times w_p$.
5: Compute the average pixel intensity in each patch $p_k \in P$ of the input image X to obtain a vector $\boldsymbol{v}_X$ (of size K).
6: Repeat step 3 for each of the gallery images to obtain a set of vectors $\boldsymbol{v}_{G_i}$, where, $i = 1, \cdots, L$. The value of L (the target gallery size) depends on the number of test images indicated in the fourth column in Table 3.1.
7: Compute the correlation between $\boldsymbol{v}_X$ and $\boldsymbol{v}_{G_i}$ corresponding to each gallery image to obtain a set of L correlation scores.
return Candidate image $X_c \in G$ that has the highest correlation, *i.e.,* $X_c = X_f$ where $f = \underset{i \in [1,\cdots,L]}{\text{argmax}} \{Corr(\boldsymbol{v}_X, \boldsymbol{v}_{G_i})\}$

we create $\boldsymbol{v}_{G_i}$ where $i = [1, \cdots, L]$, by computing the average pixel intensity in the 10 patches selected previously in the input image. Finally, we compute the correlation between the vectors $\boldsymbol{v}_X$ and $\boldsymbol{v}_{G_i}$, and select the candidate image with the maximum correlation value. The steps for selecting the candidate image are described in Algorithm 1.

After obtaining the candidate image X_c from the gallery of the target sensor S_t, the perturbations for image X are then derived with the help of X_c as described in Algorithm 2. The perturbation routine employs the following parameters: (i) α (the learning rate), (ii) η (the termination criterion), and (iii) m (the maximum number of iterations). Initially, the output perturbed image $Y^{(0)}$ is identical to the input image X. Next, we select a random patch location from $Y^{(0)}$, and create a mask matrix, $Mask$, of the same size as $Y^{(0)}$, such that the elements in $Mask$ are set to 1 for the row and column indices corresponding to the selected patch location. Then, the image $Y^{(0)}$ is perturbed iteratively using pixels from the same patch location in X_c. In each iteration, the pixels inside the selected patch are updated along two directions. The candidate image guides the direction of perturbation [123]. In the first case the perturbation is along a positive direction (implemented using line 9 in Algorithm 2), which generates Y^u. The other direction corresponds

to a negative perturbation (see line 11 in Algorithm 2), which produces Y^v. Figure 3.2(b) illustrates the role of the candidate image in the perturbation routine. Next, the noise residuals extracted from (Y^u, Y^v) are correlated with the reference pattern of the target sensor. The perturbed image yielding the maximum correlation value is then selected as the seed image for the next iteration, $iter$. This process is repeated until the relative difference between the NCC values of perturbed image Y^{iter} with respect to target sensor S_t and the original sensor S_o exceeds 10%, *i.e.* $\eta = 0.1$, or the maximum number of iterations is reached. The parameters employed in the perturbation routine are selected intuitively; for example, the learning rate is set to a small value $\alpha = 0.01$ because our objective is to perturb the image while preserving its biometric utility.

Algorithm 2: Spoofing PRNU pattern

1: **Input:** An image $X_{h\times w}$ from sensor S_o, a candidate image X_c from sensor S_t, a function $\phi(X, S_i)$ that returns the NCC value when image X is correlated with the PRNU pattern of sensor S_i ($i \in \{o, t\}$)
2: **Output:** Perturbed image Y
3: INITIALIZE: Set static parameters $\alpha = 0.01$ (learning rate), $\eta = 0.1$ (threshold), $m = 3000$ (maximum number of iterations) and $h_p = 10, w_p = 10$ (patch size), $iter = 0$ and $Y^{(0)} = X$.
4: **while** $\dfrac{\phi(Y^{(iter)}, S_t) - \phi(Y^{(iter)}, S_o)}{\phi(X, S_o)} > \eta$ **do**
5: – Choose a random patch location (p_x, p_y) from $[0, \frac{h}{h_p}]$ and $[0, \frac{w}{w_p}]$ such that $0 \le p_x < \frac{h}{h_p}, 0 \le p_y < \frac{w}{w_p}$.
6: – Construct the mask matrix $Mask$ such that $Mask[i, j] = 1$ if $\left(\lfloor \frac{i}{h_p} \rfloor, \lfloor \frac{j}{w_p} \rfloor\right) = (p_x, p_y)$, and $Mask[i, j] = 0$ elsewhere.
7: – Create a perturbed image in the positive direction $Y^u = Y^{(iter)} + \alpha Mask \odot \left(X_c - Y^{(iter)}\right)$. ▷ $\odot$ indicates element-wise product
8: – Create a perturbed image in the negative direction $Y^v = Y^{(iter)} - \alpha Mask \odot \left(X_c - Y^{(iter)}\right)$.
9: – Compute the NCC values of Y^u and Y^v for the target sensor S_t, $\phi(Y^u, S_t)$ and $\phi(Y^v, S_t)$, respectively.
10: – Set $Y^{(iter+1)} = Y^u$ if $\phi(Y^u, S_t) > \phi(Y^v, S_t)$, otherwise set $Y^{(iter+1)} = Y^v$,
11: – $iter++$,
12: – If $iter > m$ break the loop.
13: **end while**
return The final perturbed image, $Y^{(iter)}$

At the end of the routine, the perturbed image will have to be incorrectly attributed to S_t by the sensor classifier. The steps of the PRNU spoofing algorithm are illustrated in Figure 3.2(a). The

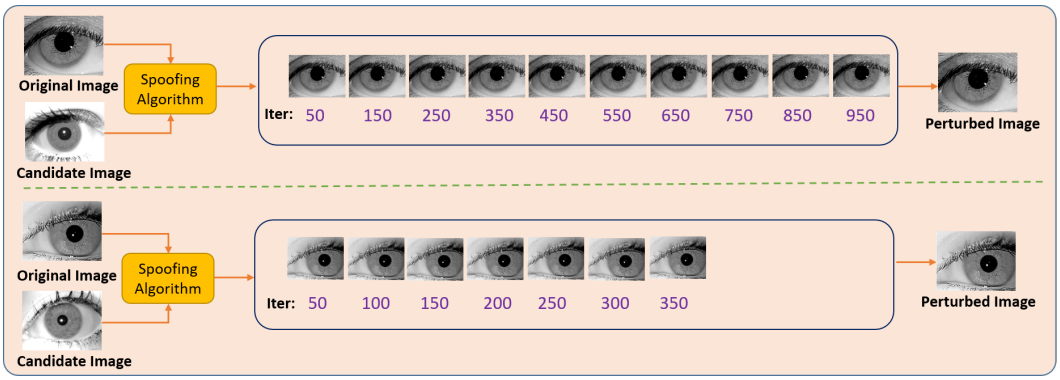


Figure 3.3: Illustration of PRNU spoofing using images belonging to the source sensor JPC and the candidate images belonging to the target sensor Aoptix.

sequence of modified images undergoing the perturbation routine is illustrated for two example iris images in Figure 3.3. perturb input images in order to spoof a PRNU-based sensor classifier, while the perturbations do not affect iris matching. As a result, after perturbations are applied to iris images, their biometric utility is still preserved.

3.2.3 Experiments and Results

In this section, we describe the datasets and sensors employed in this work, followed by the experiments conducted on the datasets. Results are reported and analyzed in the context of PRNU spoofing and iris recognition.

3.2.3.1 Datasets

Experiments are conducted using 11 different sensors from 11 iris datasets. The PRNU spoofing process typically involves a single source sensor and a single target sensor from the set of 11 sensors. The sensor details and image specifications of the 11 sensors are described in Table 3.1. Thus, there can be a total of $\lceil P_1 1 \rceil 2 = 110$ combinations for PRNU spoofing. However, for the sake of brevity, we performed 20 different PRNU spoofing experiments involving 5 sensors: $\{Aop, JPC, IC, Cog, LG40\}$. From the set of 5 sensors listed above, each sensor serves as the

source sensor while the remaining 4 sensors serve as target sensors one at a time, thus resulting in 20 different PRNU spoofing experiments.

3.2.3.2 Sensor identification before PRNU spoofing

Due to variations in image size of the source and target sensors, all images were resized to a fixed spatial resolution of 160×120 to facilitate PRNU spoofing. We then evaluated the sensor identification accuracy based on these resized images prior to PRNU spoofing. This is to determine if resizing impacts sensor identification accuracy. The sensor identification involves deriving sensor reference patterns using 55 training images, as used in [18] from each of the 11 sensors, followed by extraction of test noise residuals from images belonging to the 5 sensors, and finally correlating them. The subjects in the training set and the test set are disjoint. The sensor identification accuracy and the corresponding confusion matrix is presented in Table 3.2. The results indicate a very high sensor identification accuracy using the MLE PRNU scheme on the resized images. So we use the resized images in the experiments below.

Table 3.2: Confusion matrix for sensor identification involving unperturbed but resized images. The test noise residuals of images from 5 sensors are compared against reference patterns from 11 sensors. The last column indicates sensor identification accuracy.

Actual \ Predicted	Aop	JPC	IC	Cog	LG 40	Pan	AD	Ever	OKI	IK	LG 22	Accuracy (%)
Aop	**900**	1	2	1	9	4	3	3	9	7	1	95.74
JPC	2	**919**	4	2	5	0	0	4	1	2	1	97.77
IC	0	0	**940**	0	0	0	0	0	0	0	0	100
Cog	2	1	2	**546**	2	0	2	0	0	5	0	97.51
LG40	0	0	0	0	**529**	0	0	3	1	0	0	99.25

3.2.3.3 Sensor identification after PRNU spoofing

The PRNU spoofing process involves perturbing the original image from a source sensor using a candidate image belonging to the target sensor, whose PRNU needs to be spoofed. The impact of the perturbations on spoofing the PRNU pattern has been reported in terms of *Spoof Success Rate* (SSR), which computes the proportion of test images from the source sensor classified as belonging

to the target sensor after perturbing using Algorithm 2. The results of spoofing are presented in Table 3.3.

We implemented Baseline 1 and Baseline 2 algorithm. Baseline 2 is implemented following normalization of the source and target reference patterns with respect to the maximum intensity of the PRNU present in the two reference patterns. The normalization is required to account for the variation in the PRNU strength associated with different sensors. Ideally, the scalar terms γ and β, which serve as parameters in the baseline algorithm, need to be optimized through grid-search for a specific pair of source (S_o) and target (S_t) sensors. However, we set the scalars to a static value of 1 for two reasons: (i) for ease of computation and (ii) to provide fair comparison with the proposed algorithm which also uses fixed values of parameters for all pairs of sensors. The baseline algorithms are state-of-the art to the best of our knowledge and are, therefore, used for comparative evaluation. Examples of perturbed outputs of images spoofed using Baseline 1, Baseline 2, and the proposed algorithm are presented in Figure 3.4.

Results in Table 3.3 indicate that 15 out of 20 times the proposed algorithm outperforms Baseline 1 technique, and performs considerably better than Baseline 2 method 16 out of 20 times. The average SSR of the proposed algorithm outperforms the baseline algorithms by a significant margin. We believe that the parameters γ and β need to be tuned accurately for each pair of source and target sensors to ensure the success of the baseline algorithms. On the other hand, the proposed algorithm is successful for static parameter values: the size of patches ($h_p \times w_p$), the threshold η, the learning rate α, and the number of patches (K) (see Section 3.2.2.2). The PRNU is successfully spoofed by the proposed method in most of the cases barring the case where the target sensor is Aoptix and the source sensor is LG 4000 ($\approx$62% SSR). Inspection of the images acquired using LG 4000 sensor reveals the presence of image padding, which may negatively impact the PRNU spoofing process.

Figure 3.5 shows an input image undergoing iterative perturbations. The original (unperturbed) image belongs to the Aoptix sensor and is perturbed using a candidate image from the target sensor, Cogent. The subsequent shift of the NCC values from being the highest for the source sensor

Table 3.3: Results of PRNU spoofing where the target sensors (along the second column) are spoofed by perturbing the images from 5 source sensors, namely, Aop, JPC, IC, Cog and LG40 (along the first column). The test noise residual after the perturbation process is compared against the reference patterns of 11 sensors (see Table 3.1). The last 3 columns indicate the proportion of the perturbed images successfully classified as belonging to the target sensor and is denoted as the Spoof Success Rate (SSR). The highest values of the SSR are bolded.

Original Sensor	Target Sensor	Sensor classes compared against perturbed PRNU											SSR (%) for proposed method	SSR (%) for Baseline 1	SSR (%) for Baseline 2
		Aop	JPC	IC	Cog	LG40	Pan	AD	Ever	OKI	IK	LG22			
Aop	JPC	4	**894**	3	3	8	2	2	2	9	12	1	**95.11**	92.55	67.98
	IC	21	0	**891**	0	6	2	1	5	6	5	3	**94.79**	92.77	13.51
	Cog	7	2	3	**890**	7	5	2	2	13	5	4	**94.68**	79.89	0.21
	LG40	66	4	4	4	**836**	3	0	4	7	8	4	**88.94**	79.15	10.00
JPC	Aop	**905**	18	3	3	4	0	0	2	2	2	1	**96.28**	49.15	1.91
	IC	2	209	**712**	2	4	2	1	3	2	2	1	75.74	99.79	**100**
	Cog	3	94	4	**817**	5	5	0	5	2	1	4	**86.91**	35.53	0.21
	LG40	1	61	3	1	**861**	5	0	3	1	2	2	**91.60**	8.09	9.26
IC	Aop	**910**	0	30	0	0	0	0	0	0	0	0	**96.81**	48.72	0
	JPC	0	**797**	143	0	0	0	0	0	0	0	0	84.79	**100**	53.09
	Cog	0	0	243	**697**	0	0	0	0	0	0	0	**74.15**	46.70	0
	LG40	0	0	46	0	**894**	0	0	0	0	0	0	**95.11**	1.91	0.11
Cog	Aop	**552**	0	0	0	2	0	2	0	0	4	0	98.57	**100**	38.57
	JPC	1	**546**	0	0	1	0	2	2	0	8	0	97.50	**100**	**100**
	IC	2	0	**545**	2	2	0	2	0	1	5	1	97.32	**100**	**100**
	LG40	1	0	0	0	**550**	0	2	1	0	6	0	**98.21**	82.32	35.00
LG40	Aop	**330**	0	3	0	198	0	0	0	1	1	0	**61.91**	9.94	1.31
	JPC	0	**491**	0	0	38	0	0	2	1	1	0	**92.12**	9.38	24.20
	IC	0	0	**393**	0	136	0	0	3	1	0	0	73.73	11.44	**99.44**
	Cog	0	0	0	**479**	50	0	0	2	1	1	0	**89.87**	4.69	0.19
Average SSR (%)													**89.21**	57.60	32.75

(Aoptix) to being the highest for the target sensor (Cogent), indicates the success of the proposed method.

The average number of iterations required for successful PRNU spoofing varied between 200 to 2200. Another experiment is conducted to study the impact of increasing the number of iterations on the proposed PRNU spoofing process. This experiment is conducted for the specific case where the source sensor is LG 4000 and the target to be spoofed is the Aoptix sensor. The reason for selecting this pair is due to the poor SSR reported for this specific set of sensors (see the fifth block in Table 3.3). We speculate that with an increase in the number of iterations, the PRNU spoofing process will succeed and improve the SSR as a result. In this regard, in the new experimental set-up, the maximum number of iterations was set to 6000 (twice the earlier terminating criterion). As a result, the SSR increased considerably from 61.91% to 79.73%, *i.e.* a $\approx$ 18% increase was observed. 425 out of 533 test images belonging to the LG 4000 sensor were successfully classified

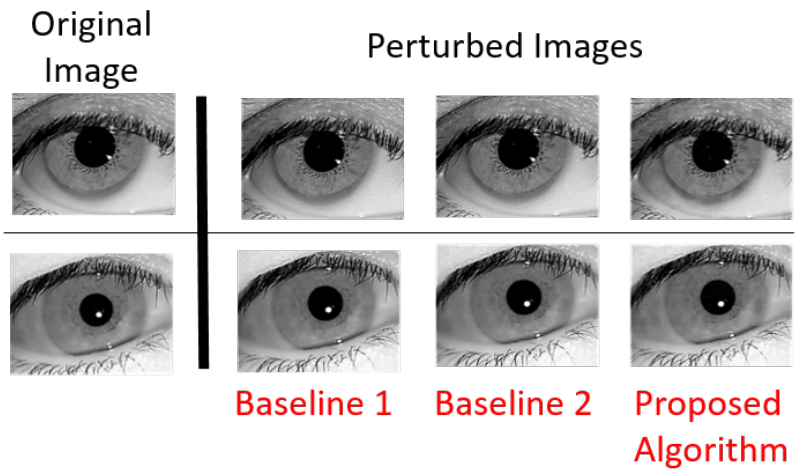


Figure 3.4: Example of PRNU spoofed images originating from the JPC 1000 sensor (first column) is illustrated for Baseline 1 (second column), Baseline 2 (third column) and the proposed method (last column). Here, the target sensor is Aoptix.

as originating from the Aoptix sensor when the number of iterations was increased.

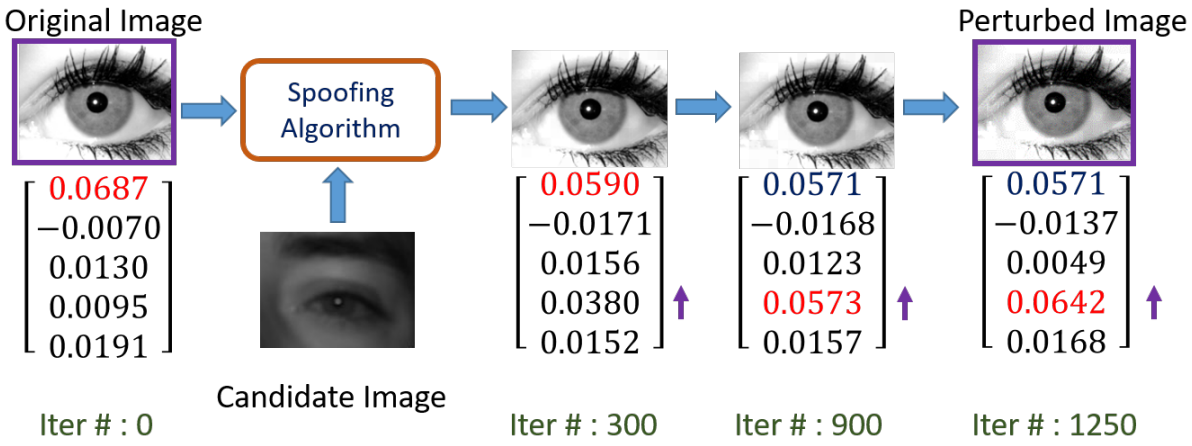


Figure 3.5: Intermediate images generated when an image from the Aoptix (S_o) sensor is perturbed using a candidate image from Cogent (S_t). For the sake of brevity, NCC values corresponding to the reference patterns of the first 5 sensors in Table 3.1 are mentioned in the figure. The arrows indicate the increase in the NCC values corresponding to the target sensor.

3.2.3.4 Retaining biometric matching utility

The impact of the perturbations on iris recognition performance is evaluated next using the VeriEye iris matcher [11]. We designed three experiments for analyzing biometric matching performance.

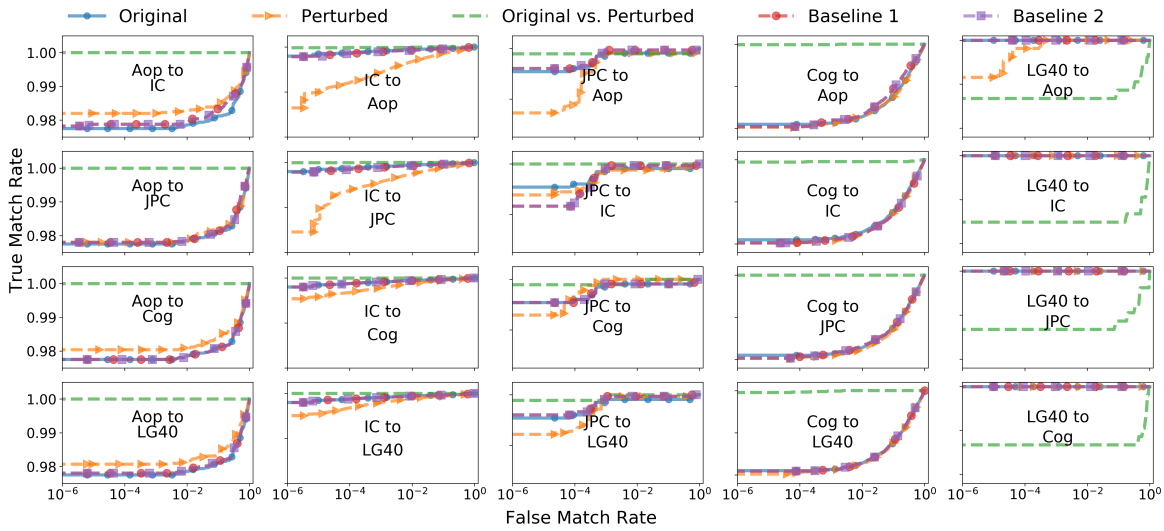


Figure 3.6: Receiver Operating Characteristics (ROC) curves of matching performance obtained using the VeriEye iris matcher software. The terms 'Original', 'Perturbed' and 'Original vs. Perturbed' indicate the three different matching scenarios (see Section 3.2.3.4). 'Original' indicates matching only unperturbed images; 'Perturbed' indicates matching only perturbed images; 'Original vs. Perturbed' indicates the cross-matching case where unperturbed images are matched against perturbed images. Note that the curves obtained from perturbed images match very closely with the curves corresponding to the unperturbed images illustrating preservation of iris recognition for each sensor depicted in each column. The results are compared with Baseline 1 and 2 algorithms discussed in Section 3.2.3.3.

First, the match scores between all pairs of iris samples before perturbation were computed. In the second experiment, we computed the match scores between all pairs of perturbed samples. In the third experiment, we computed match scores between all iris samples before perturbation and all samples after perturbation. This is referred to as the cross-matching scenario. In the third set of experiments, the genuine scores are computed by employing 2 sample images (from the same subject): one sample belonging to the set of unperturbed images and the other sample from the set of perturbed images. The impostor scores are generated by pairing samples belonging to different subjects: one image is taken from the set of unperturbed images, while the other is taken from the set of perturbed images.

Figure 3.6 shows the ROC curves obtained from these three experiments. The ROC curves confirm that the perturbed images do not negatively impact the matching utility. In the case of all the sensors, the ROC curves of the perturbed images are within a 1% deviation from the ROC

curve of the original samples before perturbation, except for the IrisCam (IC) sensor. Further, we note that the matching performance of original samples from the Cogent (Cog) sensor is degraded to begin with. We believe the reason for this degraded performance is due to the low quality of the original images. Yet, perturbations have not further deteriorated the matching performance, as evidenced by the before- and after-perturbation ROC curves that are very similar to each other.

In addition, the iris recognition performance after PRNU spoofing using the baseline algorithms is analyzed. The results indicate that the proposed method is comparable to the baseline algorithms in terms of iris recognition performance. Furthermore, we conducted a fourth experiment, where we analyzed the matching performance of those LG4000 iris images that were perturbed to spoof the Aoptix sensor after increasing the number of iterations. The result confirms that increasing the number of iterations to improve the SSR does not degrade matching performance, as is evident in Figure 3.7.

In summary, the following salient observations in the context of both PRNU spoofing and iris recognition preservation can be made.

- The PRNU pattern of a sensor can be successfully spoofed by *directly* modifying an input image, without invoking the sensor reference pattern of the target sensor. Experiments are conducted using 11 iris sensors, and the PRNU spoofing process is demonstrated using 5 sensors and compared with existing approaches. Results show that the proposed spoofing method outperforms Baseline 1 by 31.6% and Baseline 2 by 56.4% in terms of average spoof success rate.

- The proposed spoofing algorithm uses identical parameters, such as the size of patches and learning rate for all pairs of source and target sensors. This obviates the need to fine tune the method for different pairs of sensors.

- The iris recognition performance of the images perturbed using the proposed algorithm is retained within 1% of the original. This suggests the success of the proposed spoofing method in retaining the biometric utility of the modified images.

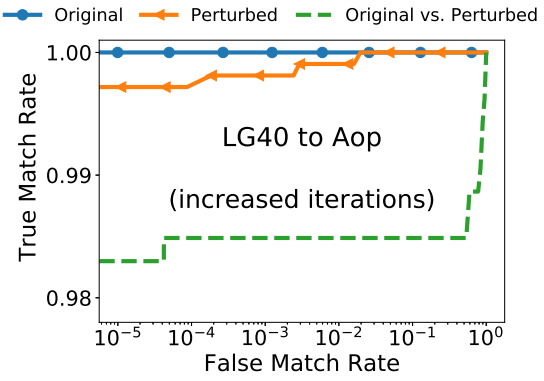


Figure 3.7: Impact of increase in the number of iterations on iris recognition performance for the pair of LG 4000 (source) and Aoptix (target) sensors.

3.2.4 Summary of the first strategy of sensor de-identification

In the first work, we design a method for PRNU spoofing that preserves biometric recognition in the context of NIR iris images. In the proposed strategy, a test image belonging to a particular sensor is modified iteratively using patches from a candidate image belonging to a target sensor, whose PRNU is to be spoofed. We examine the impact of these perturbations on PRNU spoofing as well as iris recognition performance. Experiments are conducted in this regard using 11 sensors and compared with two existing PRNU spoofing algorithms. Results show that the proposed method can successfully spoof the PRNU pattern of a target sensor and does not significantly impact the iris recognition performance in a majority of the cases.

3.3 Smartphone camera de-identification

Since smartphone devices are intricately linked to their owners, sensor identification using images from smartphone cameras can inevitably lead to person identification. This poses privacy concerns to the general populace [10] and, especially, to photojournalists [125]. *Sensor de-identification* can mitigate such concerns by removing sensor specific traces from the image. A number of sensor de-identification algorithms, particularly in the context of PRNU suppression, have been developed in the literature [64, 157]. PRNU suppression can be done by either PRNU anonymization or PRNU spoofing.

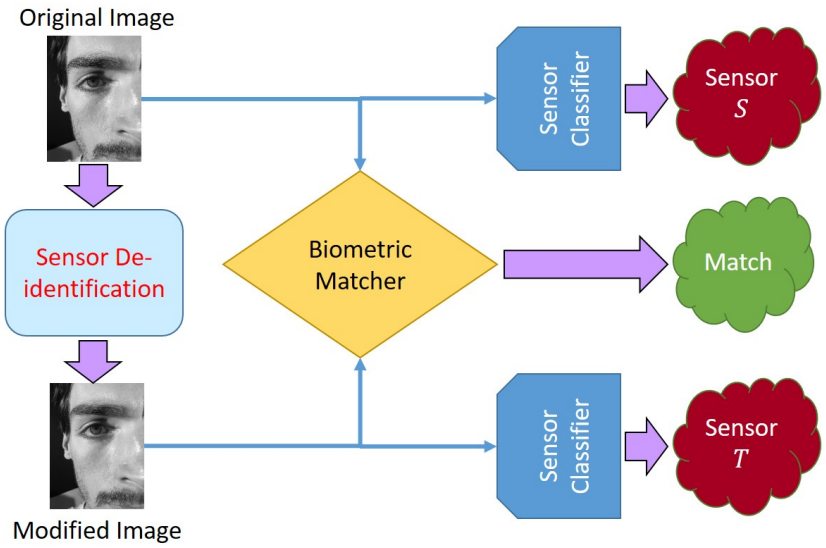


Figure 3.8: The objective of our work. The original biometric image is modified such that the *sensor classifier* associates it with a different sensor, while the *biometric matcher* successfully matches the original image with the modified image.

The objective of this work is to develop a rather simple method to perform sensor de-identification, while preserving the biometric recognition utility of the images. The key idea is illustrated in Figure 3.8. The merits of the proposed method are as follows.

1. Designing a sensor de-identification algorithm that can perform both PRNU anonymization and PRNU spoofing in a non-iterative fashion. This addresses the computational overhead incurred by the algorithms in [17, 63].

2. The proposed de-identification algorithm is applicable to different PRNU estimation schemes and works irrespective of the source and target sensors. This eliminates the need for parameter optimization and computation of the reference patterns corresponding to each pair of source and target sensors as required in [78, 107].

3. The proposed algorithm causes minimal degradation to the biometric content of the images, thus retaining their biometric utility.

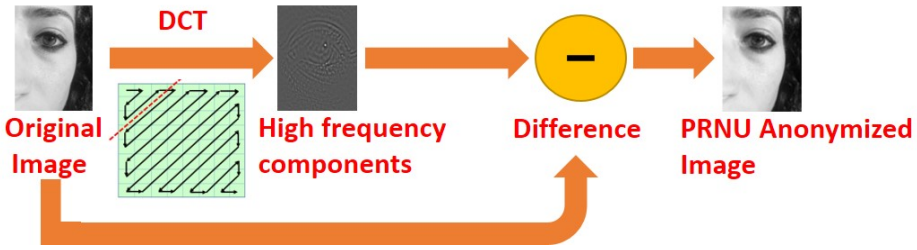


Figure 3.9: Illustration of PRNU Anonymization. The DCT coefficients are arranged such that the top-left portion has the low frequency components while the bottom-right portion encapsulates the high frequency information. The PRNU anonymized image is the result of suppression of high frequency components (see Algorithm 3, here $\eta = 0.9$).

Algorithm 3: PRNU anonymization

Input: An image $\boldsymbol{I}$ of size $h \times w$ and parameter η
Output: PRNU anonymized image $\boldsymbol{I'}$
Apply 2-dimensional DCT to $\boldsymbol{I}$; $\boldsymbol{I}_{dct} = DCT(\boldsymbol{I})$
Compute $m = \min(h, w)$ and $\alpha = round(\eta \times m)$
Extract the high frequency components as follows:
$\boldsymbol{I}_{high} = High(\boldsymbol{I}_{dct}, \alpha)$, where, the $High(\cdot, \cdot)$ operator extracts the lower triangular portion of the DCT coefficients along the anti-diagonal direction, regulated by α
Extract the low frequency components as follows: $\boldsymbol{I}_{low} = \boldsymbol{I}_{dct} - \boldsymbol{I}_{high}$
Apply inverse DCT to obtain the modified image $\boldsymbol{I'} = DCT^{-1}(\boldsymbol{I}_{low})$
return The modified image $\boldsymbol{I'}$

3.3.1 Proposed Method for smartphone camera de-identification

Discrete Cosine Transform (DCT) has been successfully used for lossy image compression [159] or for improving source camera identification [33]. The coefficients located in the top-left portion capture the low frequency components while the bottom-right coefficients encode the high frequency components. Our goal is to modify the images to perturb the PRNU pattern resulting in sensor de-identification. PRNU is a noise-like component which is dominated by the high frequency components present in an image. Thus, we propose to transform the image into the DCT domain and modulate the DCT coefficients such that the high frequency components are suppressed, while retaining the low frequency components. By suppressing the high frequency components, we mask the sensor pattern present in the image. On the other hand, we retain the low frequency components which primarily contain the scene details in the image. The scene details are pivotal for biometric recognition. Thus, we ensure preservation of the biometric utility of the image. We then apply the

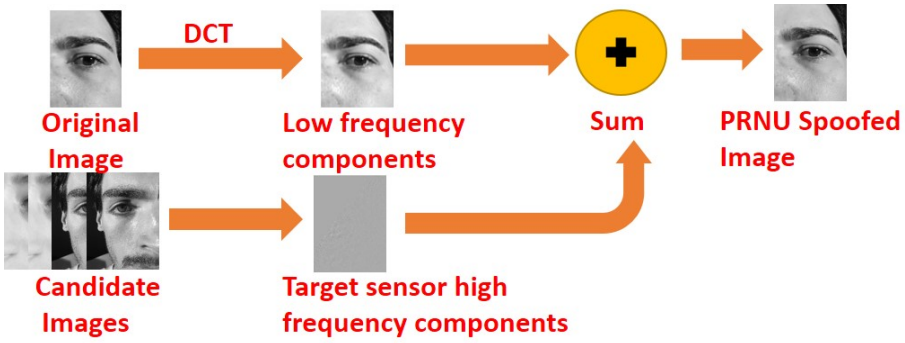


Figure 3.10: Illustration of PRNU Spoofing. The high frequency components in the original image are suppressed first, the residue being the low frequency components. The high frequency components of the target sensor are further computed from the candidate images, and added to the low frequency components of the original image, resulting in the PRNU spoofed image (see Algorithm 4, here $\eta = 0.7$).

inverse DCT, and the output is the modified image.

Algorithm 4: PRNU spoofing

Input: An image $\boldsymbol{I}$ of size $h \times w$ belonging to source sensor S, a set of N candidate images belonging to the target sensor T, where each image of size $p \times q$ is denoted as $\boldsymbol{G}^i$ $(i = [1, \cdots, N])$ and η

Output:PRNU spoofed image $\boldsymbol{I'}$

Set $i = 1$

Apply 2-dimensional DCT to $\boldsymbol{I}$, $\boldsymbol{I}_{dct} = DCT(\boldsymbol{I})$

Extract the low frequency and high frequency components, $\boldsymbol{I}_{low}$ and $\boldsymbol{I}_{high}$ as described in Algorithm 3 and set $\boldsymbol{I}_{high} = 0$

Compute $\alpha = round(\eta \times \min(p, q))$

while $i \leq N$ **do**

– Apply 2-dimensional DCT to $\boldsymbol{G}^i$, $\boldsymbol{G}^i_{dct} = DCT(\boldsymbol{G}^i)$

– Extract the high frequency components as follows: $\boldsymbol{G}^i_{high} = High(\boldsymbol{G}^i_{dct}, \alpha)$

– Apply inverse DCT to the high frequency content as follows: $\boldsymbol{G}'_i = DCT^{-1}(\boldsymbol{G}^i_{high})$

– Add the images to generate $\boldsymbol{T}_{high} += \boldsymbol{G}'_i$ and increment $i += 1$

end while

Divide by the number of images $\boldsymbol{T}_{high} = \frac{\boldsymbol{T}_{high}}{N}$

Resize $\boldsymbol{T}_{high}$ to $h \times w$ using bicubic interpolation

Apply inverse DCT to obtain the modified image $\boldsymbol{I'} = DCT^{-1}(\boldsymbol{T}_{high} + \boldsymbol{I}_{low})$

return The modified image $\boldsymbol{I'}$

To achieve sensor de-identification we perform both (i) PRNU Anonymization and (ii) PRNU Spoofing.

PRNU Anonymization: Given an image $\boldsymbol{I}$, we first subject it to DCT to yield $\boldsymbol{I_{dct}}$. We intend to suppress the high frequency information without impairing the low frequency details. To achieve this goal, we define a parameter α that serves as a regulator for high frequency suppression. α is computed as the product of the minimum of the height and width of the image $min(h, w)$, and a user-defined parameter η, rounded off to the nearest integer. All DCT coefficients present in the interval $[row = \alpha : h, col = \alpha : w]$ are set to zero. Thus, α represents the threshold for the suppression of the DCT coefficients, and that threshold is a function of the image dimensions. We discard the high frequency components and then apply inverse DCT which results in the PRNU anonymized image $\boldsymbol{I'}$. The steps are described in Algorithm 3. The process of PRNU anonymization is illustrated using an example image in Figure 3.9.

Table 3.4: Dataset specifications. The top block corresponds to MICHE-I dataset [117] and the bottom block corresponds to OULU-NPU face dataset [35]. In the MICHE-I dataset, we denote the brand Apple as 'Device 1' and the brand Samsung as 'Device 2'. Two different smartphones belonging to the same brand and model, *e.g.*, Apple iPhone5, are distinguished as 'UNIT I' and 'UNIT II'.

Smartphone Brand and Model	Device Identifier	Sensor	Image Size	Number of Images/ Number of Subjects (Training Set)	Number of Images/ Number of Subjects (Test Set)
Apple iPhone 5	Device 1	Front (F)	960×1280	55/7	344/41
	UNIT I	Rear (R)	1536×2048	55/7	355/41
Apple iPhone 5	Device 1	Front (F)	960×1280	55/6	164/20
	UNIT II	Rear (R)	2448×3264	55/6	170/20
Samsung Galaxy	Device 2	Front (F)	1080×1920	55/5	577/69
S4	UNIT I	Rear (R)	2322×4128	55/5	600/70
Samsung Galaxy S6 Edge	—	Front (F)	1080×1920	55/6	0/0
HTC Desire EYE	—	Front (F)	1080×1920	55/6	0/0
MEIZU X5	—	Front (F)	1080×1920	55/6	0/0
ASUS Zenfone Selfie	—	Front (F)	1080×1920	55/6	0/0
Sony XPERIA C5 Ultra Dual	—	Front (F)	1080×1920	55/6	0/0
Oppo N3	—	Front (F)	1080×1920	55/6	0/0
TOTAL				660/72	2,210/261

PRNU Spoofing: We want the sensor classifier to assign an image belonging to source sensor S to a specific target sensor T. To accomplish this task, we perform the following steps.

(i) First, we compute the parameter α. Next, we transform the original image $\boldsymbol{I}$ from the source sensor to the DCT domain and then extract its low frequency components (as done in Algorithm 3).

(ii) A set of N candidate images, $\boldsymbol{G}^i$, $i = [1, \cdots, N]$, belonging to the target sensor is selected, and each of them is subjected to DCT resulting in $\boldsymbol{G}^i_{dct}$ (see Section 3.3.2.2). Next, we extract the high

frequency coefficients from each $\boldsymbol{G}^{i}_{dct}$, apply inverse DCT, and then compute their average to yield $\boldsymbol{T}_{high}$. This averaged output represents the sensor traces of the target sensor.
(iii) Finally, we insert the averaged high frequency coefficients into $\boldsymbol{I}$ to generate $\boldsymbol{I'}$ which will now be classified as belonging to the target sensor, resulting in PRNU spoofing.

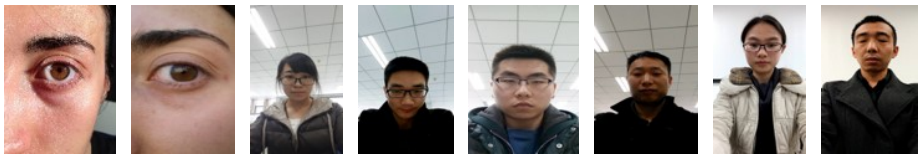
The implementation details for PRNU spoofing are described in Algorithm 4. The process of PRNU spoofing is illustrated using an example image in Figure 3.10.

3.3.2 Experiments and Results for smartphone camera de-identification

3.3.2.1 Dataset

We used the Mobile Iris Challenge Evaluation (MICHE-I) dataset [117] and the OULU-NPU face dataset [35, 103] for performing the experiments in this work. The MICHE-I dataset comprises of over 3,000 eye images from three devices: Apple iPhone 5, Samsung Galaxy S4 and Samsung Galaxy Tab 2 [71]. However, in our work, we employed the periocular images from two smartphones, Apple iPhone 5 and Samsung Galaxy S4, only. The authors in [71] discovered that two separate units of Apple iPhone 5 were used for data collection. We refer to them as Unit I and Unit II respectively. Further, the images in the dataset were acquired using the front and rear camera sensors, separately. **T**hus, the MICHE-I dataset used in this work consists of data from 6 sensors. The OULU-NPU face dataset comprises of 4,950 face videos recorded using the front cameras of six mobile devices—Samsung Galaxy S6 Edge, HTC Desire EYE, MEIZU X5, ASUS Zenfone Selfie, Sony XPERIA C5 Ultra Dual and OPPO N3. The videos were recorded in three sessions with different illumination and background scenes. **W**e only use the bonafide face videos/images in the OULU-NPU dataset corresponding to 6 sensors. See Figure 3.11. The specifications of the dataset are described in Table 3.4.

We split each dataset into a training set and a test set. We followed a subject-disjoint protocol for creating the training and test sets. The images in the training set are used for generating the reference pattern for each sensor, as indicated in the fifth column of Table 3.4. Our training set



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.11: Example images from the MICHE-I and the OULU-NPU datasets acquired using (a) Apple iPhone 5 Rear, (b) Samsung Galaxy S4 Front, (c) Samsung Galaxy S6 Edge Front, (d) HTC Desire EYE Front, (e) MEIZU X5 Front, (f) ASUS Zenfone Selfie Front, (g) Sony XPERIA C5 Ultra Dual Front and (h) OPPO N3 Front sensors.

consists of 55 images [19] from each camera sensor in the MICHE-I dataset. The OULU-NPU database contains videos, and so we selected 55 frames (20 frames from the first session, 20 frames from the second, and 15 frames from the third) from 6 subjects, for each of the 6 sensors. The test set comprises of images belonging to the MICHE-I dataset only (see the last column in Table 3.4). Thus, our dataset consists of 2,870 images corresponding to 333 subjects acquired using 12 camera sensors. Next, we describe the experiments conducted in this work.

3.3.2.2 Experimental Methodology for smartphone camera de-identification

For the *sensor de-identification* experiments, we first computed the sensor reference patterns from the traning set for each of the 12 sensors (see Table 3.4) using the three PRNU estimation schemes *viz.*, Enhanced PRNU, MLE PRNU and Phase PRNU. Next, we used a small number of images (=10) as the validation set to compute the parameter $\eta = [0, 1]$ to be used for PRNU anonymization and PRNU spoofing, separately. We estimated $\eta = 0.9$ for PRNU anonymization and $\eta = 0.7$ for PRNU spoofing. The test experiments were conducted on images belonging to the MICHE-I dataset only. **However, the evaluation process involved all the 12 sensor reference patterns.** The experiments evaluated three PRNU estimation schemes: Enhanced PRNU,[1] MLE PRNU and Phase PRNU methods. We used normalized cross-correlation for sensor identification. For the PRNU spoofing experiments, the source and target sensors were from the MICHE-I dataset and were either both front or both rear sensors. Thus, there were $2 \times {}^{[}P_3]2 = 12$ PRNU spoofing

[1] We employed Enhancement Model III and we set the user defined threshold to 6 [19, 106].

experiments. Due to the significant difference in resolutions between the front and rear sensors of smartphones, we did not perform front-to-rear or rear-to-front spoofing. We selected N, *i.e.* the number of candidate images belonging to the target sensor (see Algorithm 4), to be the number of test images for that sensor (see the last column in Table 3.4).

Table 3.5: Performance of the proposed algorithm for **PRNU Anonymization** in terms of sensor identification accuracy (%). Results are evaluated using 3 PRNU estimation schemes. 'Original' corresponds to sensor identification using images prior to perturbation. 'After' corresponds to sensor identification using images after perturbation and 'Change' indicates the difference between the 'Original' and 'After' sensor identification accuracies. A high positive value in the 'Change' field indicates successful PRNU Anonymization.

Device Identifier	**Sensors**	**Enhanced PRNU**			**MLE PRNU**			**Phase PRNU**		
		Original	After	Change	Original	After	Change	Original	After	Change
Device 1 UNIT I	Front	99.71	18.31	81.40	99.71	17.73	81.98	99.71	22.67	77.04
	Rear	99.51	16.06	83.45	97.32	16.06	81.26	98.05	21.69	76.36
Device 1 UNIT II	Front	96.34	21.34	75	96.34	25.61	70.73	93.90	26.83	67.07
	Rear	94.71	11.76	82.95	88.24	14.12	74.12	87.65	11.76	75.89
Device 2 UNIT I	Front	100	3.81	96.19	100	5.72	94.28	100	13.69	86.31
	Rear	100	4.50	95.50	100	3.17	96.83	100	6.50	93.50
AVERAGE				85.75			83.20			79.36

For the *biometric matching* experiments, we considered a periocular matcher, as many of the images used in this work are partial face images. We employed the ResNet-101 [83] architecture pre-trained on ImageNet [56] dataset for performing periocular matching. We utilized the features from layer 170 which were shown to perform the best for periocular matching in [85]. We applied Contrast Limited Adaptive Histogram Equalization (CLAHE) to the images before feeding them to the convolutional neural network. We used the cosine similarity for computing the match score between the probe and gallery images. We performed three sets of matching experiments, *viz.*, (i) original: both probe and gallery images comprise of unmodified images, (ii) after: both probe and gallery images comprise of modified images and (iii) cross: the gallery images are the original samples while the probe images are the modified images and the genuine scores are computed by utilizing 2 sample images (belonging to the same subject), *i.e.* the original image and the modified image; the impostor scores are computed by taking pairs of samples belonging to different subjects. Furthermore, we conducted experiments separately for the two acquisition settings in this database: Indoor and Outdoor.

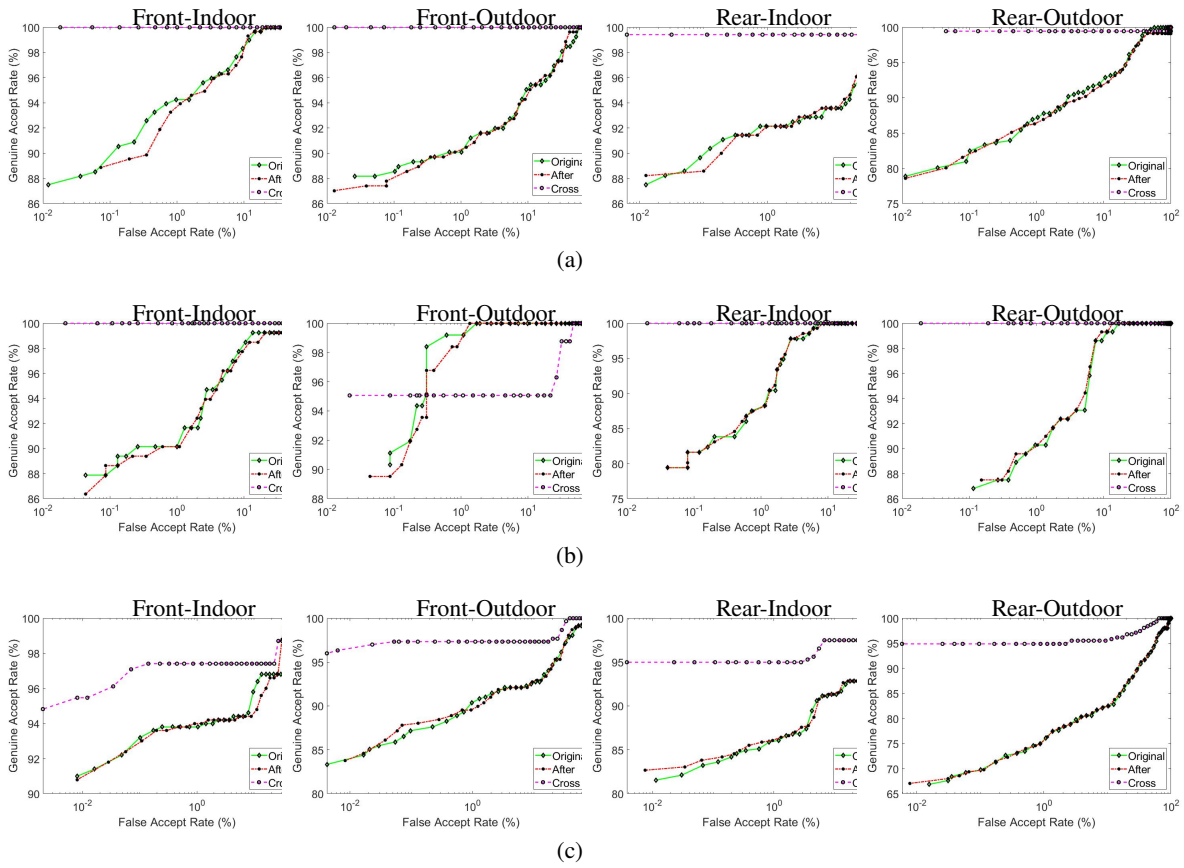


Figure 3.12: ROC curves for matching **PRNU Anonymized** images. Each row corresponds to a different device identifier: (a) Device 1 UNIT I, (b) Device 1 UNIT II and (c) Device 2 UNIT I.

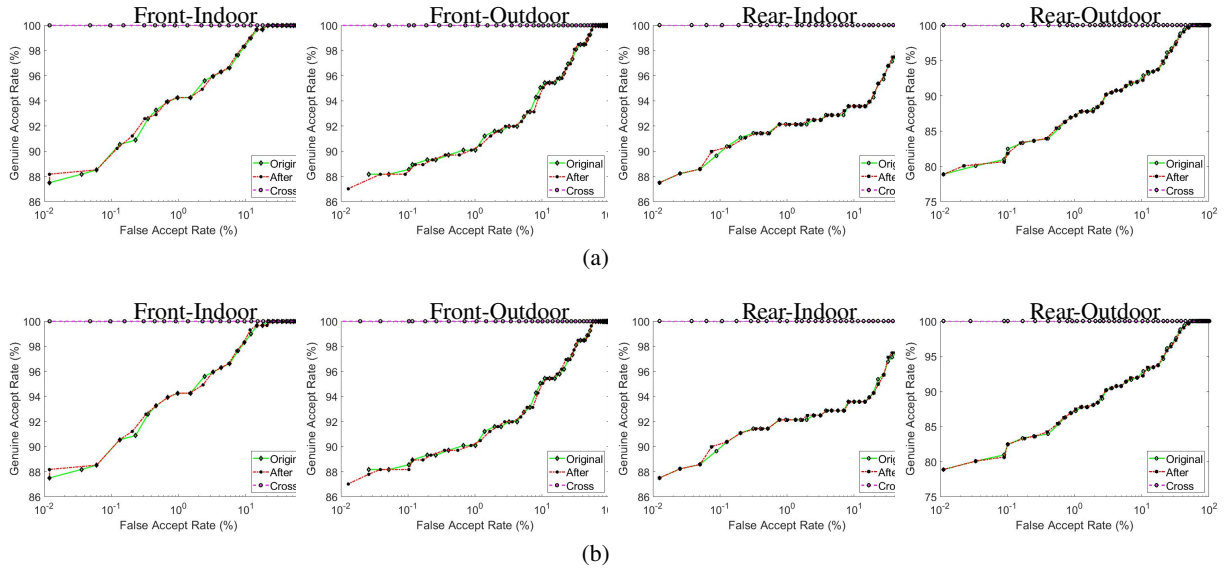


Figure 3.13: ROC curves for matching **PRNU Spoofed** images. Here, the source sensor is Device 1 UNIT I. In this case, the target sensors are: (a) Device 1 UNIT II (top row) and (b) Device 2 UNIT I (bottom row).

Table 3.6: Performance of the proposed algorithm for **PRNU Spoofing** in terms of the spoof success rate (SSR) (%). Results are evaluated using three PRNU estimation schemes. A high value of SSR indicates successful spoofing.

Source Sensor	**Target Sensor**	**Spoof Success Rate (%)**		
		Enhanced PRNU	**MLE PRNU**	**Phase PRNU**
Device 1 UNIT I FRONT	Device 1 UNIT II FRONT	100	100	100
	Device 2 UNIT I FRONT	96.80	100	100
Device 1 UNIT II FRONT	Device 1 UNIT I FRONT	100	100	100
	Device 2 UNIT I FRONT	97.56	100	100
Device 2 UNIT I FRONT	Device 1 UNIT I FRONT	100	100	100
	Device 1 UNIT II FRONT	100	100	100
Device 1 UNIT I REAR	Device 1 UNIT II REAR	100	100	94.08
	Device 2 UNIT I REAR	99.44	96.90	97.46
Device 1 UNIT II REAR	Device 1 UNIT I REAR	100	100	100
	Device 2 UNIT I REAR	100	98.82	100
Device 2 UNIT I REAR	Device 1 UNIT I REAR	100	100	100
	Device 1 UNIT II REAR	100	100	99.83
AVERAGE		99.48	99.64	99.28

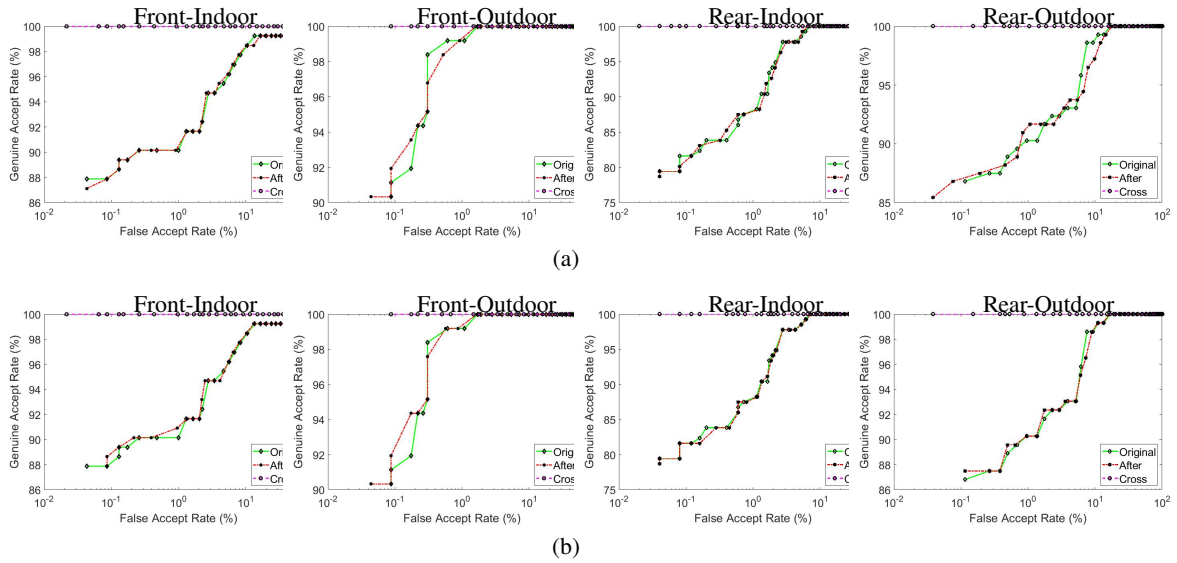
3.3.2.3 Results for smartphone camera de-identification

For the sensor de-identification experiments, we used sensor identification accuracy as the evaluation metric for PRNU anonymization and the spoof success rate (SSR) as the evaluation metric for the PRNU spoofing algorithm. For PRNU anonymization, we first compute the sensor identification accuracy of the original images. Before perturbation, the images are assigned to the correct sensor with high accuracies by all three PRNU estimation schemes (see 'Original' column in Table 3.5). Next, when the sensor classifier accepts the modified images as input, the results indicate a significant degradation in the sensor identification accuracy for a majority of the cases (see 'After' column in Table 3.5). The differences in the sensor identification accuracies before and after perturbation are reported in the 'Change' column in Table 3.5. An average difference (change) of 82.77% in the sensor identification accuracies between pre- and post-perturbed images is observed for all the three PRNU estimation schemes evaluated in this work (Enhanced PRNU: 85.75%, MLE PRNU: 83.20% and Phase PRNU: 79.36%). The results indicate successful PRNU anonymization thereby ensuring sensor de-identification.

The second set of results, pertaining to PRNU spoofing, reports the SSR for the perturbed images. SSR computes the proportion of perturbed images that are assigned to the target sensor.

The results in Table 3.6 indicate successful spoofing with respect to all the PRNU estimation schemes considered in this work. An average SSR of 99.48% is observed when evaluated using Enhanced PRNU, 99.64% when evaluated using MLE PRNU, and 99.28% when evaluated using Phase PRNU for all 12 PRNU spoofing experiments. The proposed spoofing experiment fails to confound the Phase PRNU estimation scheme, particularly when the source sensor is the rear sensor of Device 1 UNIT I and the target sensor is the rear sensor of Device 1 UNIT II. Upon analysis, we observed that the original images belonging to Device 1 UNIT II rear sensor resulted in the *lowest* sensor identification accuracy for all three PRNU estimation schemes (see Table 3.5). We speculate that the images may contain some artifacts that are interfering with reliable PRNU estimation as well as the spoofing process. Therefore, we performed another experiment where we increased the value of η from 0.7 to 0.9 for that particular spoofing experiment and we observed that the SSR increased to 100% for all 3 PRNU estimation schemes. However, visual analysis reveals that the spoofed images resulting from the two different values of η have perceptible differences ($\eta = 0.9$ results in a more blurred image than when $\eta = 0.7$ is used). Finally, we studied the performance of our PRNU spoofing algorithm when a smaller number of candidate images, N, is employed (50%, 10% and 1% of the test set). Surprisingly, even when only 1% of the test set is used as candidate images, *i.e.* $N = 4$, we observed an average SSR of 99.6% across the three PRNU estimation schemes. However, the spoofed images are significantly degraded as they contain some spurious scene details from the candidate images (possibly, the averaging operation in Step 17 of Algorithm 4 suppresses scene details more aggressively for a high value of N).

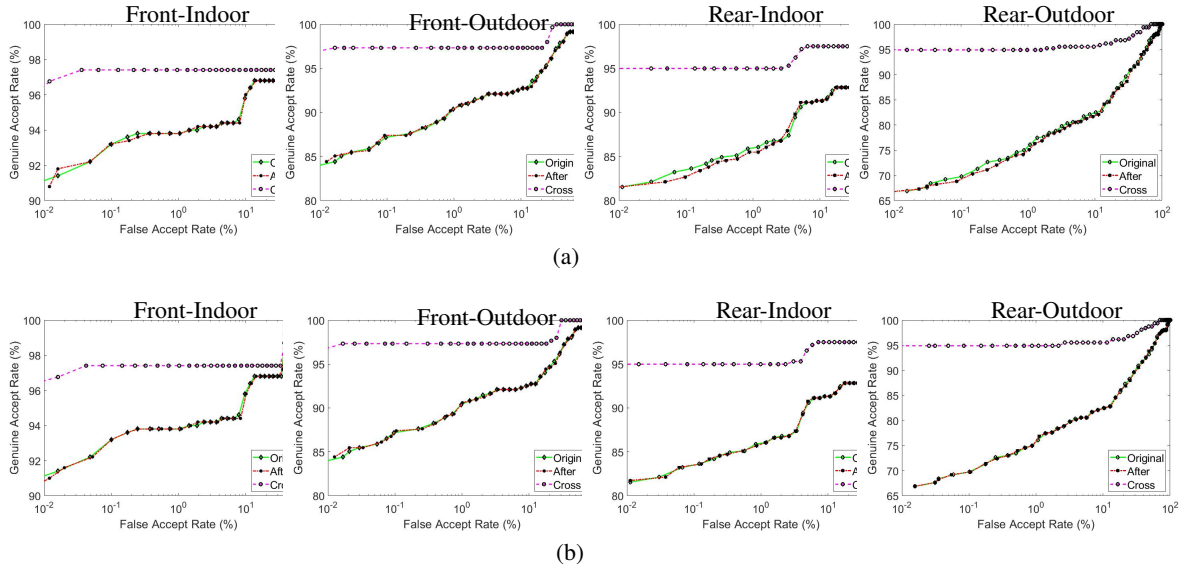
Next, we report the results for the periocular biometric recognition experiments. The periocular matching experiments indicate the preservation of the biometric utility of the images in both PRNU anonymized images and PRNU spoofed images. The ROC curves corresponding to 'Original' and 'After' matching experiments are within 1% of each other. Figure 3.12 presents the ROC curves for images subjected to PRNU anonymization. Note that Samsung Galaxy S4 results in overall lower periocular matching performance even for the original images. The cross-matching experiments result in perfect match (100%) for a majority of the cases barring the Samsung Galaxy S4 sensor.



(a)

(b)

Figure 3.14: ROC curves for matching **PRNU Spoofed** images. Here, the source sensor is Device 1 UNIT II. In this case, the target sensors are: (a) Device 1 UNIT I (top row) and (b) Device 2 UNIT I (bottom row).



(a)

(b)

Figure 3.15: ROC curves for matching **PRNU Spoofed** images. Here, the source sensor is Device 2 UNIT I. In this case, the target sensors are: (a) Device 1 UNIT I (top row) and (b) Device 1 UNIT II (bottom row).

The suppression of the high frequency components may also result in removal of edges and other details which can impact the matching performance. For the PRNU spoofing experiments, we have presented the ROC matching curves for each smartphone device or unit used in this work in Figures 3.13, 3.14 and 3.15. The matching experiments show that the perturbation scheme used for PRNU spoofing does not degrade the biometric recognition performance.

3.3.3 Summary of the second strategy for smartphone camera de-identification

In this work, we design an algorithm that perturbs a face image acquired using a smartphone camera such that (a) sensor-specific details pertaining to the smartphone camera are suppressed (sensor anonymization); (b) the sensor noise pattern of a different device is incorporated (sensor spoofing); and (c) biometric matching using the perturbed image is not affected (biometric utility). We achieve this by applying the Discrete Cosine Transform to images and further modulating the DCT coefficients to either attain PRNU anonymization or PRNU spoofing. In contrast to existing methods which involve computation of sensor reference patterns and exhaustive parameter optimization [78, 107, 154], the proposed method is simple and can achieve highly promising results. In our experiments, we considered face (partial and full) images acquired using the front and rear cameras of different smartphones resulting in data from a total of 12 camera sensors. Our proposed method results in successful camera de-identification for images without compromising the biometric matching performance. An average of $\approx$ 82.8% reduction in sensor identification is reported in the case of PRNU anonymization, and an average spoof success rate of $\approx$99.5% is observed for PRNU spoofing across the three PRNU estimation schemes evaluated in this work.

3.4 Summary

In this chapter, we proposed two different strategies for PRNU-based sensor de-identification. The motivation behind this counter-forensic measure is two-fold, firstly, to analyze the robustness of the PRNU-based sensor identification scheme, and secondly, it serves as a privacy-preserving method, particularly, for smartphone sensors. The developed strategies can achieve sensor de-

identification without compromising the biometric utility of the images. The first strategy for sensor de-identification involved an iterative patch-based image perturbation routine to spoof iris sensors. The first method outperformed state-of-the-art sensor de-identification schemes by upto 56.40%. The second strategy involved a one-shot approach that modulated discrete cosine transform coefficients to accomplish both PRNU anonymization and PRNU spoofing in the context of smartphone sensors. The proposed approach could successfully confound multiple PRNU classifiers resulting in an average reduction of 82.78% in terms of PRNU anonymization and 99.48% in terms of PRNU spoofing while retaining biometric utility within 1%.

CHAPTER 4

JOINT BIOMETRIC-SENSOR REPRESENTATION

Portions of this chapter appeared in the following publication:

S. Banerjee and A. Ross, "One Shot Representational Learning for Joint Biometric and Device Authentication," 25th International Conference on Pattern Recognition, (Milan, Italy), January 2021.

4.1 Introduction

Biometric data such as face, fingerprint or iris images reveal information about the identity of the individual as well as the identity of the device used to acquire the data [25, 135]. In some applications such as smartphone banking, it is necessary to authenticate both the *user* as well as the *device* in order to enhance security [15, 71]. This can be done by invoking two separate modules: one for biometric recognition and the other for device or sensor recognition.[1] In such cases, the system has to store two distinct templates: a biometric template denoting the identity of the user and sensor template denoting the identity of the device.

In this paper, we approach this problem by designing a *joint template* that can be used to authenticate both the user and the device simultaneously. Our objective is as follows: *Given a biometric image we would like to simultaneously recognize the individual and the acquisition device*. In the process of accomplishing this objective, we address the following questions:

1. **Why do we need to combine biometric and device recognition?**

   Smartphones are increasingly using biometrics for access control and monetary transactions. Examples include fingerprint and face recognition on iPhones and iris recognition on Samsung Galaxy S9. Device verification[2] can provide assurance that the biometric sample is being

[1] The terms "device" and "sensor" are interchangeably used in this paper. Thus, determining the identity of a smartphone camera (*i.e.,* sensor) is akin to determining the identity of the smartphone (*i.e.,* device) itself.

[2] Typically used in two factor authentication (2FA) protocol that combines any two of the three

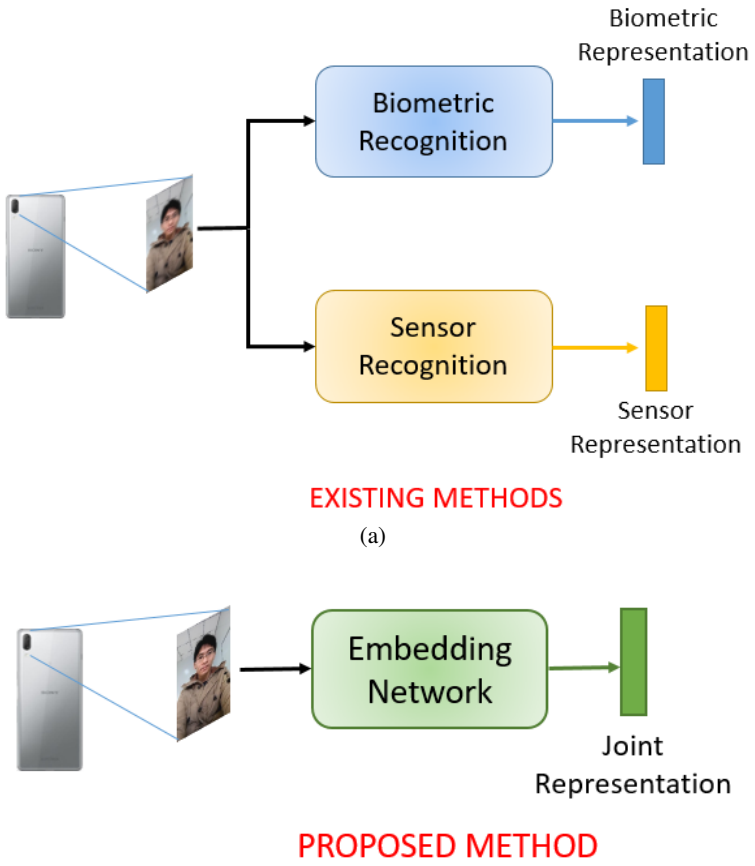


Figure 4.1: Difference between (a) methods that use separate modules for computing biometric and sensor representations, and (b) the proposed method that uses an embedding network to generate a joint biometric-sensor representation.

acquired by an authorized device. A combined biometric and device recognition system can therefore guarantee that the right person is accessing the remote service (*e.g.*, banking) using an authorized device.

2. **Can existing device verification techniques be used in the smartphone application scenario?**

   Device identification can be performed using the MAC (media access control) address, a unique networking address assigned to each device. However, in case of smartphones that have multiple network interfaces, such as Wi-Fi, 4G, bluetooth, etc., there can be multiple

factors: 'something you are' (biometrics), 'something you have' (a code on the authorized device) and 'something you know' (a password) for additional security.

MAC addresses which may be broadcasted, making them vulnerable. Alternatively, SRAM cells can be used to deduce physically unclonable cues for device identification [15]; this is a hardware-based solution and requires access to the physical device. In a mobile banking scenario, where the verification is conducted remotely, the customer provides a biometric sample in the form of an image, and some device information, but not necessarily the physical device itself. In this scenario, hardware-based solutions will be ineffective.

3. **Why do we need a joint representation?**

   Existing literature uses separate modules to tease out the biometric-specific and sensor-specific details from an image and perform feature-level or score-level fusion [15, 71]. However, they suffer from the following limitations: (i) the overall performance is limited by the weakest recognition module, and (ii) the process may not generalize well across different biometric modalities and multi-spectral sensors. Therefore, a *joint* representation that combines both biometric and sensor-specific features present in a biometric image can offer the following advantages: (i) the joint representation is not constrained by the performance of the individual recognition module, and the same *method* can be employed across different biometric modalities, and (ii) the joint representation integrates the biometric and sensor representations into a compact template, such that, the individual templates cannot be easily de-coupled; this implicitly imparts privacy to the biometric component.

4.2 Proposed Method

An image contains both low frequency and high frequency components. For example, in a face image, the low frequency components capture the illumination details while the high frequency components capture the structural details present in the face that are useful for biometric recognition. Recently, sensor recognition has been successully accomplished using Photo Response Non-Uniformity (PRNU) [112] for different types of sensors, such as DSLR sensors [42], smartphone sensors [22], and also near-infrared iris sensors [154]. PRNU is a form of sensor pattern noise in an image that manifests due to anomalies during the fabrication process and is, there-

fore, unique to each sensor. Typically, PRNU resides in the high frequencies that can be useful for sensor recognition [112]. Since the high frequencies dominate in both biometric and sensor representations, we hypothesize that there is a joint representation, that, if effectively extracted, can be utilized for both tasks of biometric and sensor recognition. Our objective is to *learn* this joint representation that lies at the intersection of the sensor and biometric space. Mathematically, it can be represented as $J(\boldsymbol{X}) = B(\boldsymbol{X}) \cap S(\boldsymbol{X})$, where $\boldsymbol{X}$ is an input biometric image, $B(\cdot)$ is the biometric representation extracted from $\boldsymbol{X}$, $S(\cdot)$ is the sensor representation computed from the same input $\boldsymbol{X}$, and $J(\cdot)$ is the joint representation. Existing methods process $\boldsymbol{X}$ using two independent routines to extract the two representations, and can optionally perform fusion, either at feature level or at score level, to make a decision. However, we propose to leverage the two representations to derive a joint representation (see Figure 4.1). The joint space can be best approximated using an embedding network that can convert images to compact representations [161]. The embedding network $\mathcal{E}$, takes two inputs, $\boldsymbol{X}$ and the dimensionality (k) of the embedding to be generated, such that $J(\boldsymbol{X}) = \mathcal{E}(\boldsymbol{X}, k) \approx B(\boldsymbol{X}) \cap S(\boldsymbol{X})$. The second argument k, allows us to regulate the dimensionality of the joint representation, which will be much lesser than the original dimensionality of the image, as well as the combined dimensionality of the two representations computed separately, *i.e.*, if $\boldsymbol{X} \in \mathbb{R}^d$, $B(\boldsymbol{X}) \in \mathbb{R}^m$ and $S(\boldsymbol{X}) \in \mathbb{R}^n$, then the joint representation $J(\boldsymbol{X}) \in \mathbb{R}^k$, where, $k << d$ and $k < (m + n)$.

In this work, we used a deep convolutional neural network that serves the role of the embedding network (see Figure 4.2). The embedding network consists of two 2-D convolutional layers and three linear layers. We used max-pooling for down-sampling the feature map and a parametric-rectified linear activation unit (PReLU) as the activation function. The embedder accepts an image, resized to 48×48, as the input and produces a 8-dimensional output, which is the *joint* representation. The choice of the dimensionality of the representation along with the experimental setup is described later (see Section 4.3.3).

The **main contributions** of this work are as follows:

1. We propose a method to learn a joint biometric and sensor representation using a one-shot

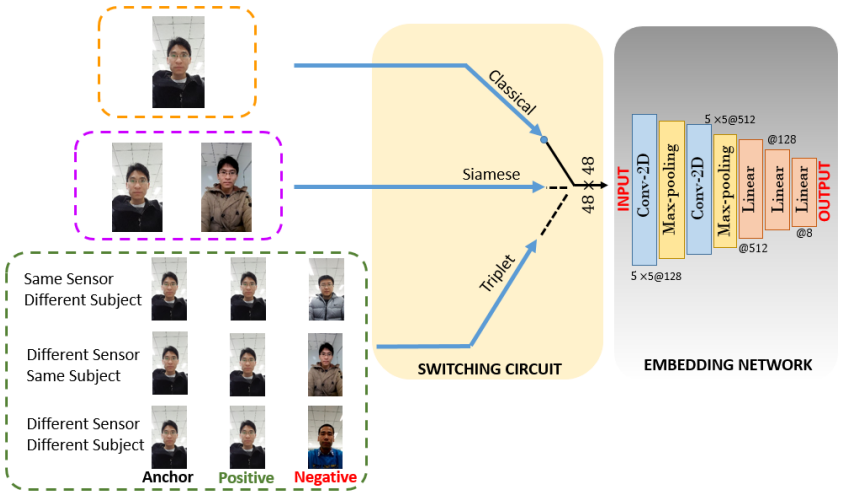


Figure 4.2: Outline of the proposed method used for computing the joint biometric and sensor representation. **Input:** A single image, or a pair of images, or 3-tuple images to the embedding network. **Output:** Joint biometric-sensor representation. The embedding network is trained in three mutually exclusive modes, *viz.,* classical mode (top row), siamese mode (middle row) and triplet mode (bottom row). The switching circuit selects only one training mode at a time.

approach that can be used in joint identification and joint verification scenarios. A correct joint identification/verification occurs *only* if both subject identity and device identity yield correct matches.

2. We employ an embedding network that can learn the joint representation irrespective of the biometric modality and the sensor used for acquisition. In this context, we evaluate the proposed method using three different biometric modalities (face, iris and periocular), and different types of sensors (iris sensors operating in the near-infrared spectrum and smartphone camera sensors operating in the visible spectrum).

3. We perform extensive experiments using different training paradigms and loss functions, and compare the proposed method with existing state-of-the-art algorithms for biometric and sensor recognition.

Table 4.1: Dataset specifications used in this work. We used three datasets corresponding to 3 biometric modalities *viz.,* iris, periocular and face. Here, we perform joint biometric and sensor recognition, so total $\#Classes$ is computed as the product of $\#Subjects$ and $\#Sensors$. (*MICHE-I dataset has a total 75 subjects, out of which the first 48 subjects were imaged using iPhone 5S UNIT I and the remaining 27 subjects were imaged using iPhone 5S UNIT II, as observed in [71]. Here, 'UNIT' refers to two different units of the same brand and model iPhone 5S, and therefore, should be treated as two different smartphones. In this case, $\#Classes$ = 375 since only a subset of the total 75 subjects were imaged using either of the two units of iPhone 5S smartphone at a time.)

Modality	**Dataset**	**Name of sensors**	**(# Subjects, # Sensors, # Classes)**	**Split**	**# Images**
Iris	CASIA-Iris V2	CASIA IrisCAM-V2, OKI IrisPass-h	(60, 2, 120)	Train Test	1,680 720
Periocular	MICHE-I	Apple iPhone5S (Front and Rear) UNITS I and II, Samsung Galaxy S4 (Front and Rear), Samsung Galaxy Tab GT2 (Front)	(75, 5 (7), 375*)	Train Test	2,278 863
Face	OULU-NPU	HTC Desire EYE, Sony XPERIA C5 Ultra Dual, MEIZU X5, Oppo N3, Samsung Galaxy S6 Edge, ASUS Zenfone Selfie	(55, 6, 330)	Train Test	5,940 2,970
TOTAL			(190, 13, 825)		14,451

4.3 Experiments

4.3.1 Datasets

In this work, we focused on three different biometric modalities, *viz.*, iris, periocular and face. To this end, we used three different datasets - (i) CASIA-Iris Image Database Version 2 [6] which contains near-infrared iris images acquired using two sensors, (ii) Mobile Iris Challenge Evaluation (MICHE-I) dataset [117] which contains partial face images acquired using two smartphones (front and rear sensors separately) and front camera of a tablet, and (iii) OULU-NPU dataset [36] which contains face images acquired using the front sensors of six smartphones. We used *only* bonafide images from the OULU-NPU dataset. Table 4.1 describes the datasets used in this work. Note that the smartphone datasets (MICHE-I and OULU-NPU) contain images acquired in the visible spectrum.

4.3.2 Evaluation Protocol

Before we describe the experiments, we present the protocol that is used to evaluate the proposed approach. We evaluate the method in two scenarios, *viz.,* (i) joint identification and (ii) joint

verification. The terms joint identification and joint verification are different from the terms used conventionally in the biometric literature. In the case of **joint identification**, a correct identification occurs *only* when both sensor and subject labels of the test sample match with the ground truth labels. To perform evaluation in the joint identification scenario, we select one embedding from each class (combines both sensor and subject label) to form the gallery, and the remaining embeddings are used as probes. We use two metrics to compute the distance or similarity between the probe and gallery embeddings and select the top three matches: (i) standardized Euclidean distance (computes the pairwise euclidean distance divided by the standard deviation) and (ii) cosine similarity. We plot the cumulative match characteristics (CMC) curves corresponding to the top three ranks. In the case of **joint verification**, two joint representations will yield a match if both the embeddings belong to the same sensor and same subject, otherwise a mismatch occurs. Incorrect match can occur in three cases as follows: (i) if the two joint representations belong to the same subject, but different sensors, (ii) if the two joint representations belong to the same sensor, but different subjects, and (iii) if the two joint representations belong to different subjects and different sensors. To perform evaluation in the joint verification scenario, we compute the distance or similarity between all the test embeddings and present receiver operating characteristics (ROC) curves to indicate the joint verification performance. We also report the true match rate (TMR) values @1% and 5% false match rates (FMR).

4.3.3 Experimental Settings

In this work, we designed the experimental settings using three different modes of training. Say, $\boldsymbol{O}$ denotes the output of an embedding network for input $\boldsymbol{X}$, *i.e.*, $\boldsymbol{O} = \mathcal{E}(\boldsymbol{X}, k)$. In the first mode, referred to as the *classical* mode, the embedding $\boldsymbol{O}$ is fed to a classification network which minimizes the cross-entropy loss computed between the ground truth label and the predicted label. The classification network in our case is a shallow network which applies PReLU activation on the embedding, followed by a fully-connected layer and then applies softmax to compute a probability value. We assigned the ground truth label for the i^{th} image, such that $l_i \in Sub_i \otimes Sen_i$, where Sub_i

denotes the subject identifier of image i, Sen_i denotes the sensor identifier for the same image and $\otimes$ denotes the tensor product. The cardinality of the set of labels $|L| = |Sub \times Sen|$. In the second mode, referred to as the *siamese* mode, a siamese network [37] is used which feeds a pair of images to the embedding network. The embedding network then computes a pair of embeddings $(\boldsymbol{O}_i, \boldsymbol{O}_j)$ and the siamese network is trained by minimizing the contrastive loss [44] computed between the pair of embeddings. We used single margin (SMCL) and double margin (DMCL) contrastive losses. Finally, in the third mode, referred to as the *triplet* mode, a triplet network [86] is trained using embeddings generated from an anchor ($\boldsymbol{O}_a$), a positive ($\boldsymbol{O}_p$) and a negative ($\boldsymbol{O}_n$) sample by minimizing the triplet loss [145]. We performed offline triplet mining as well as online triplet mining [143] with different triplet selection strategies (random negative triplet selection, semi hard negative triplet selection and hardest negative triplet selection). The triplet loss considers only one negative example at a time. Alternatively, multi-class N-pair loss function [148] considers multiple negative instances from several classes. In this work, we consider a positive example as one which belongs to the same class as the anchor (same subject and same sensor), whereas there can be three types of negative examples, *viz*., same subject but different sensor, same sensor but different subject and different sensor with different subject. Figure 4.2 illustrates the proposed method. Therefore, the number of negative classes in this work is significantly high, so we used multi-class N-pair loss using two mining techniques: (i) all positive pairs and (ii) hard negative pairs. Table 4.2 summarizes the different loss functions used in the three training modes in this work. Note that each input to the embedding network as shown in Figure 4.2 is **mutually exclusive**, *i.e.*, the embedding network can operate independently in any of the three training modes. We modified the design of an existing embedding network for implementing the different training paradigms [9]. We used learning rate = $1 \times \exp(-4)$, batch size = 4, Adam optimizer, and a step decay to reduce the learning rate by a factor $\gamma = 0.1$ every 8 epochs. The proposed network is shallow so we trained only for 50 epochs. The margin values in single margin contrastive loss and triplet losses are set to 1, while in double margin contrastive loss, both margins are set to 0.5.

For each dataset, we used a training set and a test set (see Table 4.1). The number of classes

Table 4.2: Description of the training modes and the loss functions used in this work.

Training mode	**Loss function**	
Classical	Cross entropy	
Siamese	Single margin contrastive loss (SMCL) Double margin contrastive loss (DMCL)	
Triplet	Offline triplet mining	
	Online triplet mining	Random negative Semi-hard negative Hardest negative
	Multi-class N-pair	All positive pair Hard negative pair

is computed as the product of the number of sensors and number of subjects in that dataset. For example, CASIA-Iris V2 dataset has 60 subjects and 2 sensors, so total number of classes is $60 \times 2 = 120$. Each class has 20 images, therefore, the total number of images (samples) is 2,400 (20×120). The training and test partitions follows a 70:30 split. So, for a single class, out of 20 samples, 14 samples are randomly selected as the training set and the remaining 6 samples form the test set. Similar protocol is followed for the remaining datasets.

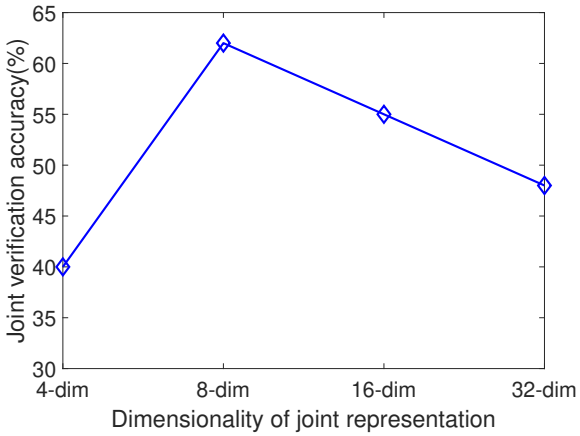


Figure 4.3: Variation in the joint verification performance as a function of the dimensionality of the joint representation. Experiment is conducted on the validation set using 50 images from the MICHE-I dataset and four dimensionality values *viz.,* $\{4, 8, 16, 32\}$. 8-dimensional embedding resulted in the highest joint verification accuracy, and is therefore selected in this work.

Next, in the training phase, the embedding network accepts an image (resized to 48×48) as input. Different image resolutions were used $\{28 \times 28, 48 \times 48, 96 \times 96\}$, but 48×48 provided

optimal trade-off between accuracy and training time. The embeddings are trained in (i) classical, (ii) siamese and (iii) triplet modes. Then, in the testing phase, we computed the embeddings from the test set. We evaluate the test embeddings in joint identification and joint verification scenarios.

Although deep learning-based sensor identification methods exist in the literature [14, 69, 116], we used Enhanced PRNU [106] (with enhancement Model III) as the sensor identification baseline for all three modalities due to its low computational burden and effectiveness against multi-spectral images [19]. Enhanced PRNU requires creation of sensor reference patterns, that serve as gallery and test (probe) noise residuals, that are correlated with the reference patterns. We used training images to compute the sensor reference patterns and test images for correlation. A test image is assigned to the sensor class resulting in the highest correlation value. See [22] for more details. Test noise residuals computed from JPEG images can be matched successfully against sensor reference patterns computed from RAW images [155], thereby, justifying the use of PRNU as a state-of-the-art sensor identification baseline. We used COTS matcher as the biometric recognition baseline for iris and face modalities. For the periocular modality, we used a pretrained ResNet-101 architecture [85] and used the features from layer 170 as the biometric representation for the test samples. This particular architecture is used because it has demonstrated good performance in biometric verification on the MICHE-I dataset [22]. The gallery comprises the training images and the probes are the test images. Since, PRNU can only be used for the task of sensor identification, we selected to implement both the baselines only in identification scenario.

We further conducted an experiment using a validation set comprising 50 images from the MICHE-I dataset (excluded from the test set) to analyze the effect of the dimensionality of the embedding on the verification performance. To this end, we used four values of $k = \{4, 8, 16, 32\}$, and then selected that value which results in the highest performance for the remaining experiments.

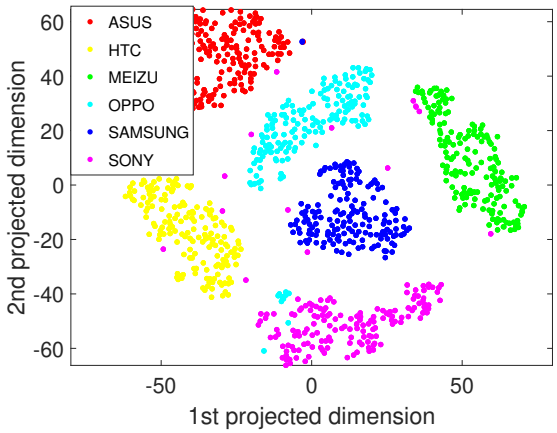


Figure 4.4: 2-D projection of the embeddings using t-SNE used for sensor identification in the OULU-NPU dataset. Each sensor class is sufficiently discriminated from the rest of the sensors.

4.4 Results and Analysis

4.4.1 Selection of the metric and dimensionality of embedding

In terms of the choice of the distance/similarity **metric**, we observed that standardized euclidean distance metric resulted in better performance compared to the cosine similarity metric. This can be attributed to the standardization process which takes into account the intra-class and inter-class variations in the embeddings. In terms of the choice of the **dimensionality** of the embedding, we observed that 8 ais the optimal value, since, it resulted in the best performance (64 on the MICHE-I validation set) as indicated in Figure 4.3. Therefore, we used 8-dimensional embedding and standardized Euclidean distance metric for all the experiments. Furthermore, we presented the t-SNE [156] visualization of the performance of the embedding network in terms of sensor identification for the OULU-NPU dataset in Figure 4.4. The well-separable clusters corresponding to the six sensors demonstrate the capability of the embedding network used in this work.

4.4.2 Performance of each of the three training modes

In terms of **training algorithms**, the overall results in both joint identification and joint verification scenarios indicate that the embedding network trained in *siamese* mode outperformed the remaining

Table 4.3: Results in the joint identification scenario. Results are reported in terms of Rank 1 identification accuracies (%). A correct joint identification implies that *both* sensor and subject resulted in a match. Mismatch of either subject or sensor or both will result in an incorrect joint identification.

Dataset	Method for baseline	Baseline performance			Proposed method (%)
		Sensor identification (%)	Biometric identification (%)	Joint identification (%)	
CASIA-Iris V2	PRNU COTS	100.00	56.52	56.52	**89.67**
MICHE-I	PRNU ResNet-101	99.86	18.05	18.05	**47.53**
OULU-NPU	PRNU COTS	98.48	84.24	83.13	**99.81**

training paradigms (see Figures 4.5 and 4.6). The reason for the superior performance of siamese network can be attributed to the use of contrastive loss. Out of the two contrastive losses, single margin contrastive loss outperformed double margin contrastive loss. The contrastive loss considers a pair of embeddings at a time, and tries to either minimize the distance between them if they belong to the same class, or increases the distance between them by some margin if they belong to different classes. On the other hand, triplet loss tries to *simultaneously* minimize the distance between the anchor and positive sample, whereas, maximize the distance between the anchor and negative samples. In this work, the number of negative classes is very high (in a 330 class dataset, 1 class is positive and the remaining 329 classes are negative). This makes the task of triplet loss much more complex as compared to contrastive loss. Given the huge variation in the possible combination of negative triplets (see Figure 4.2), we suspect that the triplet loss struggled to determine the accurate decision boundary between the positive and negative classes, resulting in an overall reduction in performance. We investigated different types of triplet mining strategies, and observed that online triplet mining outperformed offline triplet mining and multi-class N-pair in a majority of the cases.

4.4.3 Results of the joint identification experiment

In terms of the performance in **joint identification scenario**, Table 4.3 compares the results with the baseline performance for all the datasets. We reported the baselines for sensor identification

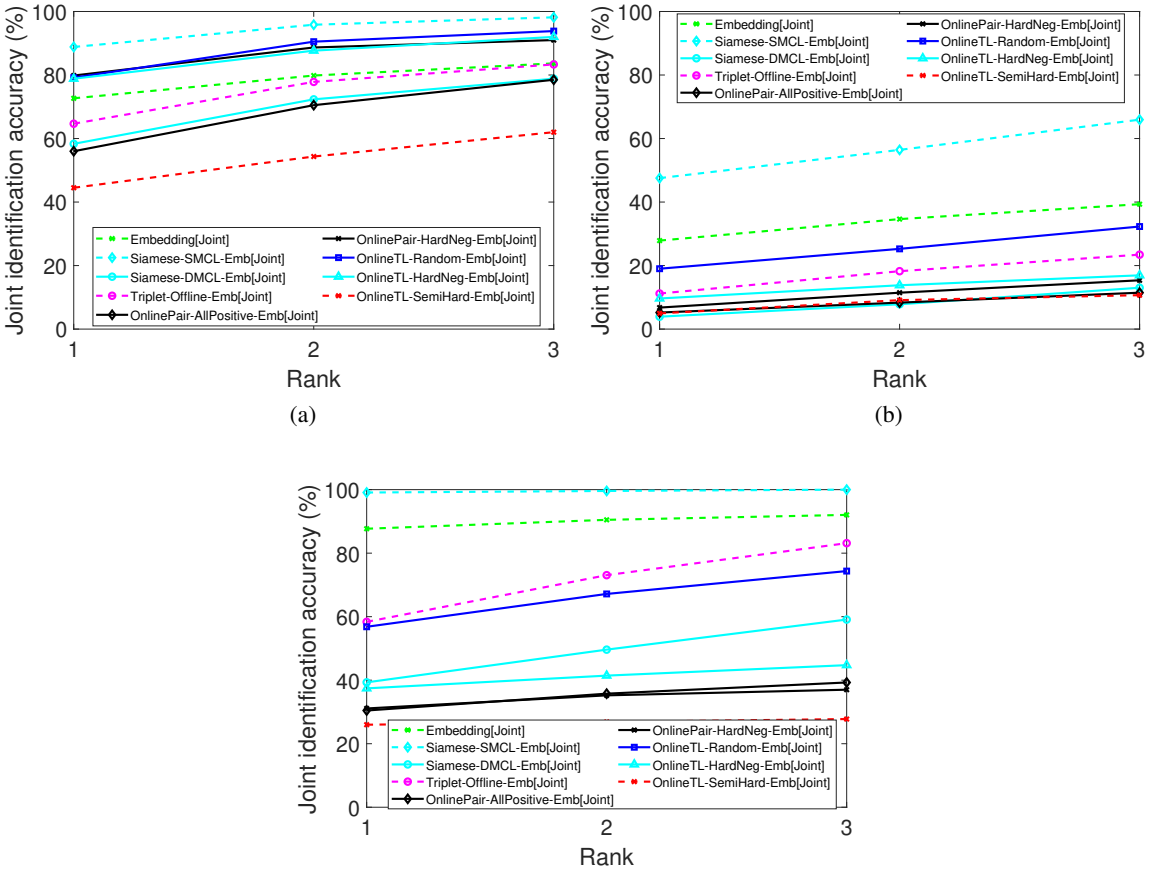
Table 4.4: Results in the joint verification scenario. Results are reported in terms of true match rate (TMR) at false match rates (FMRs) of 1% and 5%.

Dataset	TMR@FMR=1%	TMR@FMR=5%
CASIA-Iris V2	90.00	98.00
MICHE-I	62.00	90.00
OULU-NPU	100.00	100.00

(PRNU), biometric identification (COTS or ResNet), followed by joint identification, separately. We reiterate that joint identification involves a correct match only if both sensor and subject labels are correct to allow fair comparison with the proposed method. Results indicate that the proposed method outperformed the baseline (joint identification) by 26.41% averaged across all three datasets computed at Rank 1. The poor performance for the MICHE-I dataset can be attributed to two factors - firstly, the large number of classes (= 375) compared to rest of the datasets (see Table 4.1), and secondly, the diverse acquisition settings (indoor vs. outdoor) resulting in degraded biometric recognition, and subsequently leading to overall poor performance. Surprisingly, the proposed method can still outperform the baseline by ∼30%. We have further analyzed this performance in Section 4.4.5. CMC curves indicate the superior performance of the siamese network in contrast to classical and triplet networks.

4.4.4 Results of the joint verification experiment

In terms of the performance in **joint verification scenario**, Table 4.4 reports the results. Results indicate that the proposed method achieved an average joint TMR of 84% @1% FMR, and an average TMR of 96% @5% FMR, indicating the strong representative capability of the joint representation. ROC curves in Figure 4.6 indicate that the joint representation learnt using siamese network trained with single margin contrastive loss (see the curve marked Siamese-SMCL-Emb[Joint]) outperformed the remaining joint representations. We would like to point out that in [71], the authors achieved 23% (by using feature level fusion) and 86 (by using score level fusion) at 5% FMR on the MICHE-I dataset (the authors excluded the Samsung Galaxy Tab 2 subset of the



(a) (b)

(c)

Figure 4.5: Cumulative Matching Characteristics (CMC) curves for the proposed method in the **joint identification scenario** for the following datasets used in this work: (a) CASIA-Iris V2 (b) MICHE-I and (c) OULU-NPU. Refer to Table 4.2 for the different training networks and loss functions indicated in the legend in an identical order.

MICHE-I dataset, which we included in our evaluations). Although their objectives were different compared to the proposed work (they adopted a fusion rule for integrating their proposed biometric and sensor recognition performances), we would like to indicate that the task of joint recognition is difficult. In spite of that, the proposed method performed reasonably well.

4.4.5 Analysis of the performance of the proposed method on MICHE-I dataset

In both cases of joint identification and joint verification experiments, we observed that the performance of the proposed method evaluated on the MICHE-I dataset was relatively worse compared

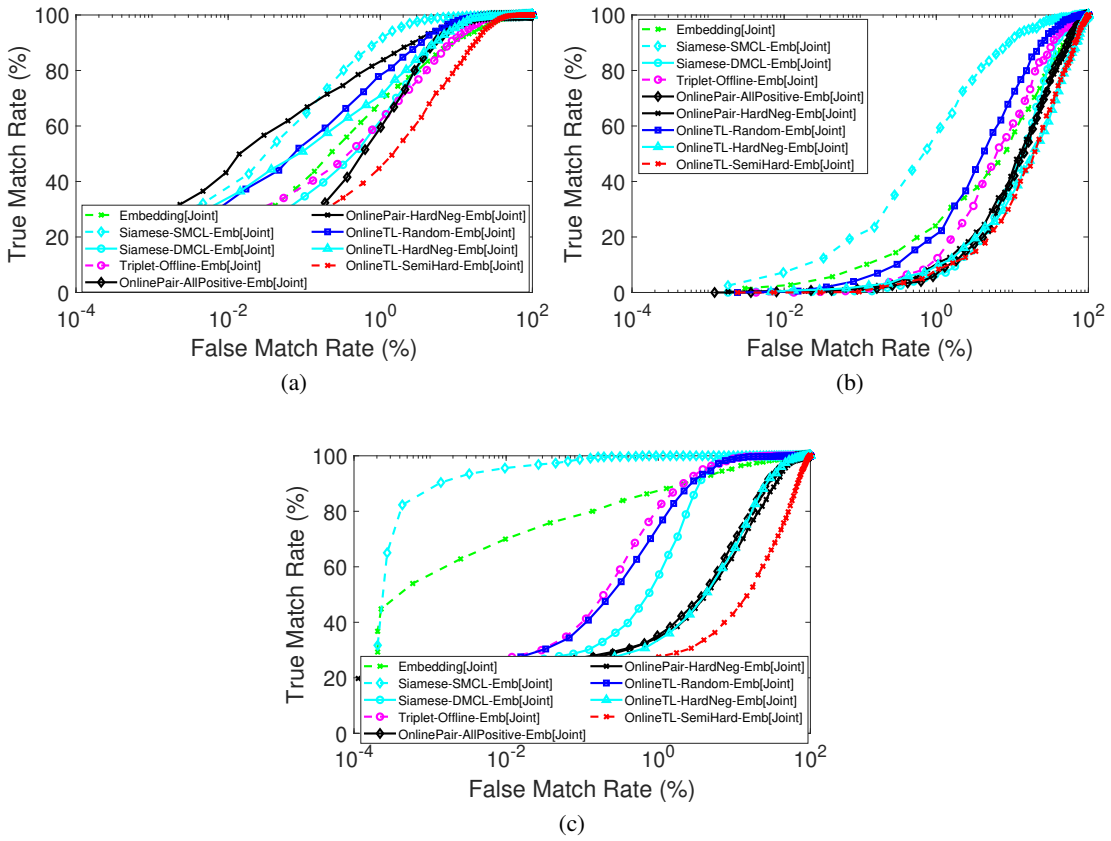
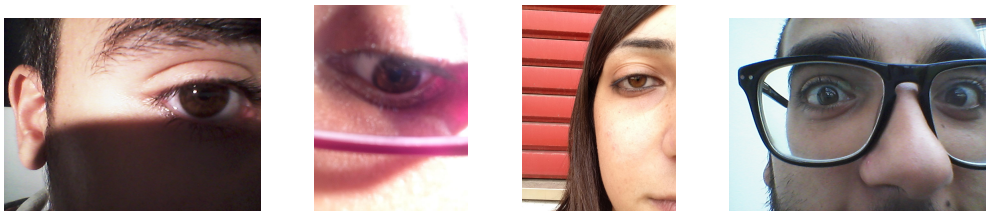


Figure 4.6: Receiver Operating Characteristics (ROC) curves for the proposed method in the **joint verification scenario** for the following datasets used in this work: (a) CASIA-Iris V2 (b) MICHE-I and (c) OULU-NPU. Refer to Table 4.2 for the different training networks and loss functions indicated in the legend in an identical order.

to the remaining two datasets. We posit that the poor performance can be attributed to two reasons: (i) the image characteristics, and (ii) the variation in the performance across different lateralities, *i.e.,* left vs. right periocular images. MICHE-I dataset was assembled as a part of an iris challenge evaluation and contains images acquired in unconstrained settings (indoor and outdoor settings) having occlusions (specular reflection and downward gaze). See some challenging images from the MICHE-I dataset images in Figure 4.7. In contrast, CASIA and OULU datasets contain images acquired in controlled settings.

We presented the CMC curves corresponding to joint identification results for two lateralities separately in Figure 4.8. Results indicate that the proposed method performed better on left periocular images compared to right periocular images. This variation in the performance across



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

Figure 4.7: Example images from the challenging MICHE-I dataset. (a) Occlusion, (b) Downward gaze and specular reflection, (c) Prominent background in the outdoor setting and (d) A single image containing both eyes but labeled as right eye image (061_GT2_OU_F_RI_01_3 where, RI indicates right eye).

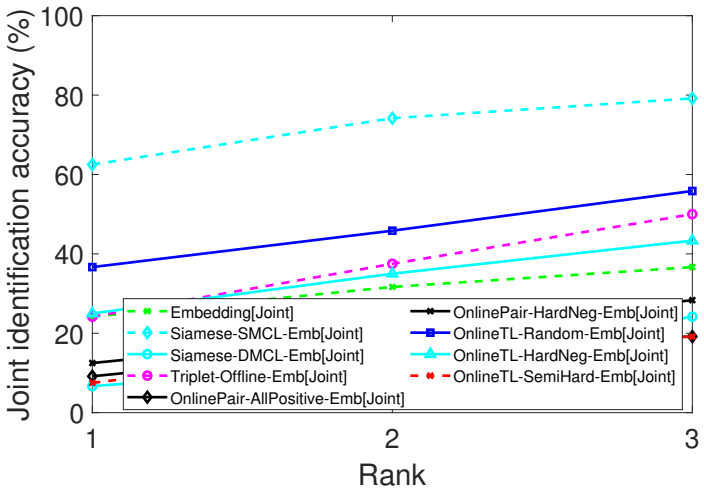
the two lateralities resulted in the overall poor performance on the entire MICHE dataset. MICHE dataset has an imbalanced distribution of lateralities, only 30% of the total number of subjects contain left periocular images. We hypothesize that the imbalanced distribution coupled with some mislabeled test case (see Figure 4.7(d)) may have further compounded the challenges, resulting in an overall poor performance.

The **main findings** from the experiments are as follows:

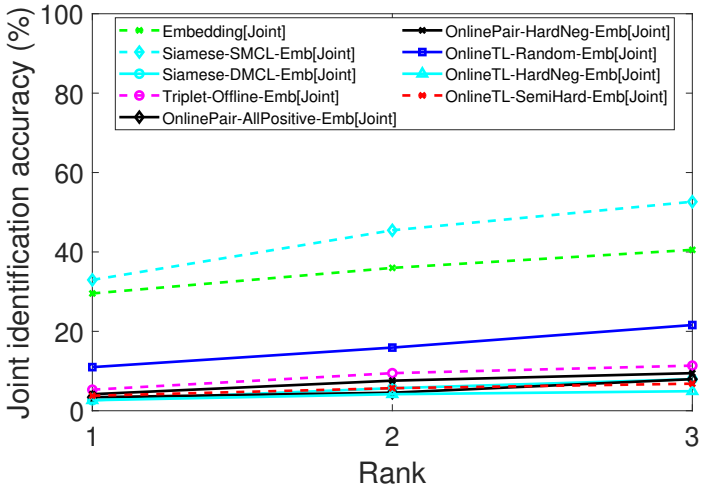
1. The joint biometric and sensor representation performed well in both joint identification scenario, with an average identification accuracy of ~80 computed at Rank 1, and an average joint verification accuracy of 96% at a false match rate of 5%, averaged across the three biometric modalities.

2. The representation is *robust* across three modalities (iris, face and periocular), and different sensors (near-infrared iris sensors and visible smartphone sensors).

3. The joint embedding outperformed baselines that used state-of-the-art commercial biometric matchers and sensor identification schemes across three datasets corresponding to three biometric modalities and multi-spectral sensors.



(a)



(b)

Figure 4.8: Cumulative Matching Characteristics (CMC) curves for the proposed method in the joint identification scenario for the MICHE-I dataset evaluated separately on the two lateralities, *i.e.*, on the (a) Left periocular images and on the (b) Right periocular images. Results indicate that the proposed method performs better on the left periocular images compared to the right periocular images.

4.5 Summary

In this chapter, we proposed a one-shot method to simultaneously authenticate the user and the device from a single image, say a face or an iris image. To accomplish this task, we developed a method to learn a joint representation that can be used for combined biometric and sensor (device) recognition. The joint representation can be used in remote application scenarios (remote bank-

ing) that employ multiple factor authentication. Additionally, the joint representation is implicitly privacy-preserving as the biometric and sensor representations cannot be trivially separated. We evaluated the proposed approach on multiple datasets belonging to three different biometric modalities (iris, face and periocular) in both joint identification and joint verification scenarios. We observed the best performing results of joint identification accuracy of 99.81% at Rank 1 and a joint verification accuracy *i.e.,* True Match Rate of 100% at 1 False Match Rate using the proposed method on face images.

CHAPTER 5

IMAGE PHYLOGENY TREE FOR NEAR-DUPLICATE BIOMETRIC IMAGES

Portions of this chapter appeared in the following publication:

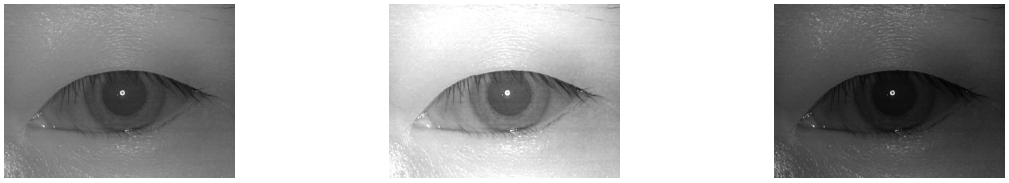
S. Banerjee and A. Ross, "Computing an Image Phylogeny tree from Photometrically Modified Iris Images," 3rd International Joint Conference on Biometrics, (Denver, USA), October 2017.

5.1 Introduction

In the previous chapters, we focused on the sensor-based forensic analysis of biometric images. To this end we presented methods that can be used for identifying biometric sensors, followed by sensor de-identification. From this chapter onward, we will delve into the content-based forensic analysis of biometric images. We will focus on the particular problem of image phylogeny in the context of biometric images. We will primarily explore face and iris modalities, but we first begin with the iris modality in this chapter.

The performance of a biometric recognition system, say an iris recognition algorithm naturally depends on the quality of the iris image. A photometrically modified iris image may adversely affect the iris recognition performance [136]. An iris image may be subjected to a sequence of photometric transformations such as brightening, contrast adjustment and gamma correction, resulting in a family of transformed images, all of which are directly or indirectly related to the original image. Example of such photometric transformations are presented in Figure 5.1.

In this work, we explore the feasibility of determining the relationship between a set of photometrically modified NIR iris images using image forensic principles. In some applications, it may be necessary to automatically deduce the relationship between such transformed images in order to determine the structure of image evolution [60] and to represent it in the form of an Image Phylogeny Tree (IPT). An IPT is a tree-like structure depicting the hierarchical relationship between a family of transformed images. Each image is represented as a node, and an edge exists between a pair of nodes if one image is derived from the other. The source image is then termed as the parent



(a) (b) (c)

Figure 5.1: Example of photometric transformations applied to an NIR iris image. (a) Original image, (b) brightness adjusted image, and (c) contrast adjusted image.

node and the transformed image is referred to as the child node. The relationship between the child and parent nodes can be modeled using a parametric function.

In this work, our objective is as follows. For a given set of photometrically transformed near-duplicate iris images, (a) determine the transformation parameters between pairs of images in the set and (b) represent the relationship between the related images in the form of an IPT. We assume that no prior knowledge about the transformations applied to the images is available. The first objective is achieved by exploring the use of three models for modeling the transformation between pairs of images: (a) global linear model; (b) local linear model; and (c) global quadratic model. The second objective is achieved by using the estimated parameters of the transformation between pairs of images, along with a variant of the Oriented Kruskal algorithm [61], to compute a Directed Acyclic Graph (DAG) that is used to represent the IPT (IPT-DAG). Further, we conduct experiments to validate the relevance of the three models in representing some popular photometric transformations. The principal contributions of our paper are as follows: (a) employing three parametric models, namely, global linear, global quadratic and local linear models to obtain the *best fit* to some photometric transformations such as, brightness adjustment, Gaussian smoothing, contrast limited adaptive histogram equalization (CLAHE), median filtering and gamma correction; (b) estimating the parameters of the three models, and further using the estimated parameters to compute an asymmetric dissimilarity measure for constructing the IPT-DAG; and (c) evaluating the performance of the proposed method in terms of IPT reconstruction accuracies for different tree configurations.

5.2 Proposed Approach

Consider a family of photometrically transformed images denoted by the set $\{\mathbf{I}_1, \cdots, \mathbf{I}_N\}$. Our objective is to construct an IPT as shown in Figure 5.2. This entails computing an asymmetric dissimilarity measure between every image pair $(\mathbf{I}_i, \mathbf{I}_j)$, where, $i, j = 1, \cdots, N$. The dissimilarity measure is computed from the parameters $\boldsymbol{\theta}$ of a transformation model, $T(\mathbf{I}|\boldsymbol{\theta})$, that relates $\mathbf{I}_i$ and $\mathbf{I}_j$. The pipeline of the proposed approach as presented in Figure 5.2 can be broadly classified into two steps. The first step constitutes the parameter estimation process for parameterized transformation models. The second step is the construction of the IPT-DAG using the results from the first step.

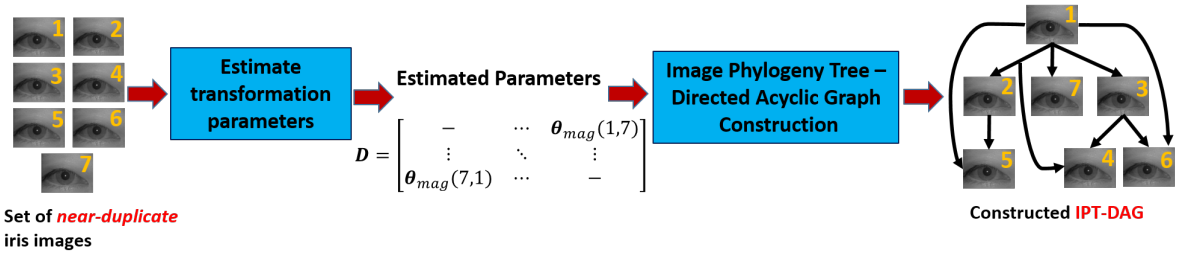


Figure 5.2: General framework for parameter estimation and IPT reconstruction from a set of near-duplicate and related iris images.

5.2.1 Parametric Transformations

In order to estimate the transformation between a pair of images, three different parametric models are considered: (a) global linear (GL) model, (b) local linear (LL) model, and (c) global quadratic (GQ) model. The global models assume the application of the same set of transformation parameters to every pixel. On the other hand, the local model assumes that the image is tessellated into several non-overlapping patches, and each patch is subjected to a different set of transformation parameters (see Figure 5.3). Different parametric models are considered in this work because some commonly used image enhancement and denoising operations are global (e.g., brightening) while others are applied in patches (e.g., CLAHE).

Next, we will discuss the three different models, the parameters associated with each of these models and their respective parameter estimation processes.

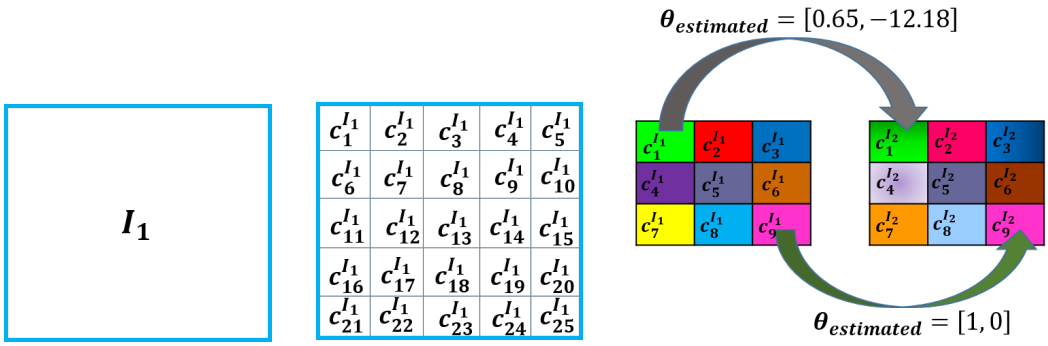


Figure 5.3: Illustration of global model optimization vs. local model optimization. (a) Global model optimizes with respect to entire image, (b) local model optimizes with respect to each of the tessellated patches in the image, and (c) local optimization for a pair of tessellated images.

5.2.1.1 Global Linear (GL) Model

The GL transformation model is denoted in Eqn. (5.2.1), where a represents the multiplicative coefficient and b represents the bias or the offset term.

$$T(\mathbf{I}|a,b) = a\mathbf{I} + b. \tag{5.2.1}$$

The parameters for the global linear model are $\boldsymbol{\theta}_{GL} = [a,b]^T$. We will describe the inverse compositional (IC) update rule [16, 26, 111] for estimating the parameters of the GL photometric model. For an image pair $(\mathbf{I}_i, \mathbf{I}_j)$, where, $\mathbf{I}_i$ is the source image and $\mathbf{I}_j$ is the target image, we consider that $\mathbf{I}_j$ has been subjected to some photometric transformation operation, $T(\mathbf{I}_j|\boldsymbol{\theta})$. We want to estimate the parameters of the transformation model, $\boldsymbol{\theta}$, such that it results in minimum photometric error (PE) between $T(\mathbf{I}_j|\boldsymbol{\theta})$ and $\mathbf{I}_i$.

The parameter estimation involves minimizing the PE between the image pair using gradient descent. The PE serves as the vector valued objective function formulated as,

$$PE = \min_{\delta\boldsymbol{\theta}} \|\mathbf{I}_i - T(\mathbf{I}_j|\boldsymbol{\theta})\|_2{}^2. \tag{5.2.2}$$

The objective function is minimized in an iterative procedure with respect to the incremental parameters $\delta\boldsymbol{\theta}$ resulting in the optimal parameter $\boldsymbol{\theta}_{opt}$. Eqn. (5.2.2) requires minimization to be performed in the photometric space of the target image. The solution to Eqn. (5.2.2) involves computation of the gradient which needs to be performed in every iteration, making the procedure computationally intensive. An elegant solution to this problem can be obtained by minimizing the objective function in the photometric space of the source image. To achieve this, the IC update rule [26] is employed, which can be written as,

$$T(\cdot|\boldsymbol{\theta}) = \hat{T}^{-1}(T(\cdot|\boldsymbol{\theta})|\delta\boldsymbol{\theta}), \tag{5.2.3}$$

where, $\hat{T}(\cdot|\delta\boldsymbol{\theta})$ is an incremental transformation with respect to parameter $\delta\boldsymbol{\theta}$. The primary objective is to express the updated transformation in terms of the composition of the current transformation, $T(\cdot|\boldsymbol{\theta})$, and the inverse of the incremental transformation $\hat{T}^{-1}(\cdot|\delta\boldsymbol{\theta})$. After applying the IC update rule to Eqn. (5.2.2), the least squares problem can be finally written as, $\min_{\delta\boldsymbol{\theta}} \|\hat{T}(\mathbf{I}_i|\delta\boldsymbol{\theta}) - T(\mathbf{I}_j|\boldsymbol{\theta})\|_2^2$.
Let, $\mathbf{W}_j = T(\mathbf{I}_j|\boldsymbol{\theta})$. Thus, the equation becomes,

$$\min_{\delta\boldsymbol{\theta}} \|\hat{T}(\mathbf{I}_i|\delta\boldsymbol{\theta}) - \mathbf{W}_j\|_2^2. \tag{5.2.4}$$

The solution for the above expression requires the use of first order Taylor's expansion which introduces the Jacobian of the photometrically transformed source image, $\hat{T}(\mathbf{I}_i|\delta\boldsymbol{\theta})$. Applying Taylor's expansion to Eqn. (5.2.4) results in
$\min_{\delta\boldsymbol{\theta}} \|\mathbf{I}_i + \mathbf{J}_\mathbf{I}^i \cdot \delta\boldsymbol{\theta} - \mathbf{W}_j\|_2^2 = \min_{\delta\boldsymbol{\theta}} \|\mathbf{J}_\mathbf{I}^i \cdot \delta\boldsymbol{\theta} - \mathbf{E}\|_2^2$.

Here, $\mathbf{J}_\mathbf{I}^i$ represents the Jacobian matrix, and the difference vector $\mathbf{E}$, also known as the error image, indicates $\mathbf{W}_j - \mathbf{I}_i$. For the GL model, the Jacobian is computed as,

$\mathbf{J}_\mathbf{I}^i = [\frac{\partial T(\mathbf{I}_i|a,b)}{\partial a}, \frac{\partial T(\mathbf{I}_i|a,b)}{\partial b}] = [\mathbf{I}\mathbf{v}_i, \mathbf{1}]$.

Here, $\mathbf{1}$ is a column vector of all ones of the same length as $\mathbf{I}\mathbf{v}_i$, which is the vectorized form of the image $\mathbf{I}_i$, such that $\mathbf{I}\mathbf{v}_i = vect(\mathbf{I}_i)$. Thus, the Jacobian depends only on the source image ($\mathbf{I}_i$) pixels. The incremental parameter vector is solved using the following equation:

$$\delta\boldsymbol{\theta} = (\mathbf{J}_\mathbf{I}^{iT}\mathbf{J}_\mathbf{I}^i)^{-1}\mathbf{J}_\mathbf{I}^{iT}\mathbf{E}. \tag{5.2.5}$$

The optimal parameters, $\boldsymbol{\theta}_{opt} \leftarrow [a_{opt}, b_{opt}]^T$ are computed using the inverse update rule as follows :

$$a_{opt} \leftarrow \left[\frac{a_{old}}{1+\delta\boldsymbol{\theta}_a}\right]; b_{opt} \leftarrow \left[\frac{b_{old}-\delta\boldsymbol{\theta}_b}{1+\delta\boldsymbol{\theta}_a}\right]. \tag{5.2.6}$$

5.2.1.2 Local Linear (LL) Model

Some photometric transformations (e.g., median filtering) are applied to images in patches thus producing non-uniform changes throughout the image. A global model fails to capture such local variations. An effective solution is to apply a different transformation at each local region in an image. The process can be simplified by tessellating the image into several non-overlapping patches and applying the GL based parameter estimation process iteratively on each of these patches. Intuitively, the LL model seeks local optimization as opposed to global optimization guaranteed by the GL model. For estimating transformation parameters between a pair of images, we assume that the images are in geometric correspondence with respect to each other.

For the LL model, consider the image $\mathbf{I}$ to be tessellated into m equal-sized patches $\{\mathbf{c}_1^{\mathbf{I}}, \cdots, \mathbf{c}_m^{\mathbf{I}}\}$, where, $\mathbf{c}_1^{\mathbf{I}}$ denotes the first patch of image $\mathbf{I}$. Thus, the transformation for each patch can be represented as,

$$T(\mathbf{c}_i^{\mathbf{I}}|a_i, b_i) = a_i\mathbf{c}_i^{\mathbf{I}} + b_i. \tag{5.2.7}$$

For the i^{th} patch, the transformation parameters are $\boldsymbol{\theta}_i = [a_i, b_i]$. The parameter estimation process for each pair of patches is identical to the GL model based approach. Upon estimation of the parameters for each patch, the optimal transformation parameters for the entire image is computed as their average. The aggregation is necessary due to the patch based approach adopted in the LL model. Each patch is essentially a matrix of pixel intensity values and some of these patches have low condition number, *i.e.,* they are close to being singular matrices, thus, making parameter estimation for these patches unreliable. However, averaging the estimated parameters reduces the effect of the ill-conditioned patches and estimates the optimal parameter fairly accurately. The solution can be expressed as follows, $\boldsymbol{\theta}_{LL} = \left[\frac{\sum_{i=1}^{m} a_i}{m}, \frac{\sum_{i=1}^{m} b_i}{m}\right]^T$.

5.2.1.3 Global Quadratic (GQ) Model

Image filtering operations are widely used to enhance iris images, which aid in iris segmentation and subsequent iris normalization. However, such operations cannot be approximated using a simple linear model. A classical example is Gaussian smoothing which is used extensively for the purposes of image denoising. A simple quadratic model may better model the non-linearities inherent to such transformations compared to simple linear models. Thus, the third model considered in this work is the global quadratic model denoted as,

$$T(\mathbf{I}|a,b,c) = a\mathbf{I}^2 + b\mathbf{I} + c, \tag{5.2.8}$$

where, a, b and c represent the scalar coefficients of the transformation.

The parameters for the quadratic model are $\boldsymbol{\theta}_{GQ} = [a,b,c]^T$. The least squares estimation (LSE) technique can be used for computing the coefficients of the quadratic model. Eqn. (5.2.8) can be rewritten as,

$$T(\mathbf{I}|a,b,c) = [abc] \begin{bmatrix} \mathbf{I}^2 \\ \mathbf{I} \\ \mathbf{1} \end{bmatrix}. \tag{5.2.9}$$

Eqn. (5.2.9) can be simplified by considering the following notations: $\mathbf{t} = vect(T(\mathbf{I}|a,b,c))$, $\varphi(\mathbf{I}) = vect\left(\begin{bmatrix} \mathbf{I}^2 \\ \mathbf{I} \\ \mathbf{1} \end{bmatrix}\right)$. Substituting the above notations in Eqn. (5.2.9) results in

$$\mathbf{t} = \boldsymbol{\theta}^T \cdot \varphi(\mathbf{I}) = \varphi(\mathbf{I})^T \cdot \boldsymbol{\theta}. \tag{5.2.10}$$

Since, the output, $\mathbf{t}$, can be expressed as a weighted linear combination of the input, $\varphi(\mathbf{I})$, it is linear in terms of the the parameters, $\boldsymbol{\theta}$. Thus, it can be solved using linear or ordinary LSE and has a closed form solution. The solution for the parameters (*i.e.,* the coefficients of the quadratic model) can be expressed as:

Algorithm 5: Asymmetric dissimilarity measure computation

Input: An image pair, $(\boldsymbol{I}_i, \boldsymbol{I}_j)$
Output: Dissimilarity matrix, $\boldsymbol{D}$
Normalize the source image $(\boldsymbol{I}_i)$ and the target image $(\boldsymbol{I}_j)$ by dividing both the images by the maximum pixel intensity value in $\boldsymbol{I}_i$
initialization:
    $\boldsymbol{\theta} \leftarrow [1, 0]^T, threshold \leftarrow norm(\boldsymbol{\theta}), iter \leftarrow 0, maxIter \leftarrow 100;$
pre-computation:
    $\boldsymbol{Jacob} \leftarrow [normalized_{\boldsymbol{I}_i}, \mathbf{1}];$
    $\boldsymbol{Hessian} \leftarrow \boldsymbol{Jacob}^T * \boldsymbol{Jacob};$ ▷ * indicates matrix multiplication operation
loop:
while $threshold < 1 \times 10^{-8} \quad OR \quad iter \leq maxIter$ **do**
    $transformed_{img} \leftarrow \boldsymbol{\theta}(1). * normalized_{\mathbf{I}_j} + \boldsymbol{\theta}(2)$ ▷ .* indicates element-wise multiplication operation
    $Error_{img} \leftarrow transformed_{img} - normalized_{\mathbf{I}_i}$
    $\delta\boldsymbol{\theta} \leftarrow \sum_{rows} \boldsymbol{Jacob}^T * Error_{img}$
    $\boldsymbol{\theta} \leftarrow \left[\frac{\boldsymbol{\theta}(1)}{1+\delta\boldsymbol{\theta}(1)}, \frac{\boldsymbol{\theta}(2)-\delta\boldsymbol{\theta}(2)}{1+\delta\boldsymbol{\theta}(1)}\right]^T$
    $threshold \leftarrow norm(\delta\boldsymbol{\theta})$
    $iter \leftarrow iter + 1$
end while
return $\boldsymbol{D}(\boldsymbol{I}_i, \boldsymbol{I}_j) \leftarrow norm(\boldsymbol{\theta}); Photometric_{error} \leftarrow norm(Error_{img})$

$$\boldsymbol{\theta}_{opt} = \left(\varphi(\mathbf{I})\varphi^T(\mathbf{I})\right)^{-1} \varphi(\mathbf{I})\mathbf{t}. \tag{5.2.11}$$

In summary, for all three models, the transformation parameters are estimated in both directions: from the first image to the second image ($\mathbf{I}_i \rightarrow \mathbf{I}_j$) and, also, from the second image to the first image ($\mathbf{I}_j \rightarrow \mathbf{I}_i$). The magnitude of the parameters are asymmetric in the two directions. The magnitude of the estimated parameters, computed as $L2$-norm of the vector $\boldsymbol{\theta}$, is then used to compute the $N \times N$ dissimilarity matrix, $\mathbf{D} = [\boldsymbol{\theta}_{magnitude}(i, j)]_{i,j=1}^{N}$, which quantifies the dissimilarity between every pair of images in the input set. The process of parameter estimation (for the GL model) and dissimilarity measure computation is summarized in Algorithm 5.

5.2.2 IPT-DAG Construction

The magnitudes of the estimated transformation parameters between every pair of images serve as the elements of the dissimilarity matrix required by the IPT-DAG construction algorithm. The IPT construction algorithm as described in [61] assumes that each node has a single parent and constructs a minimal spanning tree (MST) from the dissimilarity matrix. The Oriented Kruskal algorithm constructs the MST by first sorting all the elements of the dissimilarity matrix, and then creating edge between nodes, say, i and j, directed from $j \rightarrow i$, such that i is the parent node and j is the child node. The edge is created *only* if both the nodes do not belong to the same tree, and if node j has not been assigned a parent. However, a dissimilarity matrix, which is unable to successfully discriminate between the source (node i) and the target (node j) image, may misclassify i as the child node of j. There is no corrective procedure to amend the reconstruction, since the local relationships are not examined by the algorithm. Consequently, this will negatively impact the IPT reconstruction accuracy. The authors in [57] have used optimum branching algorithm to remedy the above situation, which assumes an initial root node and iteratively tries different root nodes to arrive at the optimal solution; but this leads to higher algorithmic complexity. We propose relaxing the MST construction by considering the IPT as a directed acyclic graph (DAG). Our objective is to have a single attempt at IPT reconstruction, where reconstruction involves no prior knowledge about the correct root, and at the same time be able to evaluate the reconstruction accuracy using a single criterion. As such, converting both the original tree and the reconstructed tree into their respective DAG forms may also aid in better understanding the relationships between the nodes, *i.e.,* the images within a set.

The IPT-DAG construction algorithm begins by sorting the elements of the dissimilarity matrix with respect to each node at a time. This allows searching for local relationships in contrast to global relationships as is the case with the Oriented Kruskal algorithm. In every iteration, a row of the dissimilarity matrix is selected (the row index corresponding to a single node, say, $node_1$) and a set of potential candidates is determined from the remaining nodes $\{node_2, \cdots, node_N\}$. The potential nodes are the vertices which will possibly share an edge with the current node under

consideration. A node (e.g., $node_k$, $k = 2, \cdots, N$) is considered to be a potential candidate if the magnitude of the estimated transformation parameter between the pair $(node_1, node_k)$ is less than 5 times the minimum value of that of all the elements belonging to the current row (row_1) of the dissimilarity matrix under consideration. Once the potential candidates are selected, the direction of the edge is decided by comparing the parameter magnitudes in the forward direction and the reverse direction. A lower magnitude will result in an edge in the corresponding direction. The output of the algorithm is a data structure with two columns: the first column named $Child$ contains the child nodes and the second column named, $Parent$ comprises of the corresponding parent nodes. Algorithm 6 describes the steps in the IPT-DAG construction process. In our approach, we only reconstruct a single tree for a given set of images.

Algorithm 6: IPT-DAG construction

Input: Dissimilarity matrix **D** of size $n \times n$
Output: IPT-DAG containing n nodes
for each row in **D do**
    $currentnode \leftarrow rowindex$
    sort row in ascending order,
    $l \in sorted - row$;
    $minval \leftarrow minimum(sorted - row)$;
    check ▷ Determine the potential nodes sharing edge with row
    if $(l < 5 \times minval)$ **then**
        $potential - nodes \leftarrow l$
    end if
    for each potential node, m **do** ▷ Determine the direction of the edge
        if $\mathbf{D}(currentnode, m) < \mathbf{D}(m, currentnode)$ **then**
            $Child \leftarrow m$;
            $Parent \leftarrow currentnode$.
        else
            $Child \leftarrow currentnode$;
            $tParent \leftarrow m$.
        end if
    end for
end for
return IPT-DAG $\leftarrow [Child, Parent]$

5.2.3 Performance Evaluation

The reconstructed IPT-DAG is compared against the original IPT which is also converted to a DAG by inserting links between each node and its *ancestor* (if not currently present). The accuracy is computed as follows,

$$Accuracy = \frac{Originaledges \cap Reconstructededges}{Cardinality\ of\ the\ set\ of\ Originaledges}, \tag{5.2.12}$$

where, $Reconstructededges = \{Parent \rightarrow Child\}$. The IPT-DAG structure indicates the root node, the leaf nodes, ancestral relationships and edges. The node appearing most frequently in the $Parent$ column is interpreted as the Root node and is connected to all the remaining nodes in the reconstructed IPT-DAG. Similarly, the leaf nodes appear only in the $Child$ column of the data structure obtained as output of the IPT-DAG construction algorithm (discussed in Section 3.2), and nodes appearing in both columns ($Child$ and $Parent$ columns) represent the intermediate nodes.

5.3 Experiments

In this section, the datasets and the experimental protocols used in this work are described. Results are reported in terms of (a) photometric error for the parameter estimation algorithms of the three parametric transformation models, and (b) the IPT reconstruction accuracy.

5.3.1 Datasets and Experimental Methodology

The three sets of experiments conducted in this work are described next.

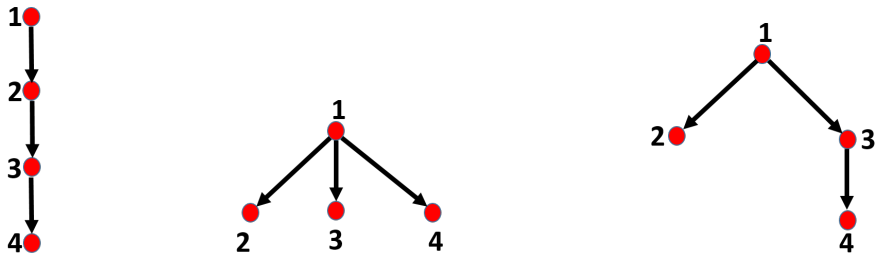
In the first set of experiments, 300 iris images from the CASIAv2 Device2 dataset [6] were subjected to 5 popular photometric transformations with varying parameters. The photometric transformations and the range of the parameters associated with each transformation, are described in Table 5.1. For each pair of original and transformed images, the GL, GQ and LL models are used to estimate the parameters of the transformation. For the LL model, each image is tessellated into non-overlapping patches of size 16×16. A total of 300 transformed pairs are obtained. The goal of this experiment is to demonstrate that some popular photometric transformations can be

reasonably approximated by one or more of these 3 parametric models. The final photometric error (formulated in Step 18 of Algorithm 5) obtained after convergence of the estimation algorithm is computed for each of the three models, and the model yielding the lowest error is declared as the best fit for that specific photometric transformation.

Table 5.1: Photometric transformations and selected range of parameters used for the first set of experiments.

Photometric Transformation	**Parameters**	**Range**
Brightness adjustment	[a,b]	$a \in [0, 10], b \in [-30, 30]$
Median	[m,n]	$m \in [3, 20], n \in [3, 20]$
CLAHE	contrast limit, size of window	$c \in [0, 0.09], m \times n \in [5, 8] \times [5, 8]$
Gaussian smoothing	standard deviation	$stddev \in [2, 8]$
Gamma correction	gamma	$gamma \in [0.1, 2]$

In the second set of experiments, 1200 images from the CASIAv2 Device2 dataset and 1992 images from the CASIAv4 Thousand [7] dataset were assembled together to form our experimental dataset resulting in a total of 3192 images. *Each* image of the dataset is subjected to a sequence of photometric transformations resulting in IPT of different configurations. The parameters for the second experimental protocol are presented in Table 5.2. The different tree configurations analyzed in this paper are presented in Figure 5.4. A total of 3192 IPTs were constructed for each of the three configurations. Parameter estimation is conducted independently on each set of transformed images.



(a) (b) (c)

Figure 5.4: Examples of image phylogeny tree configurations considered in this work. (a) Breadth= 1, Depth= 3, (b) Breadth= 3, Depth= 1 and (c) Breadth= 2, Depth= 2 .

In the third set of experiments, the 3192 images from the second experiment are subjected to all 5 transformations resulting in a single complex IPT of breadth 3 and depth 2. The resulting IPT

and an example iris image undergoing multiple transformations are exhibited in Figure 5.5. For this experiment, 3192 trees were constructed and evaluated in terms of reconstruction accuracy. In practice, we cannot guarantee that all the images belonging to a particular tree will be subjected to the same transformation. Some images may arise due to application of global operations, while others may be a consequence of local operations. Accurate reconstruction of such IPTs is of practical importance. The primary objective of the third set of experiments is to evaluate which of the three parametric functions (GL, GQ and LL models) will be well suited for modeling an IPT generated using sequences of multiple photometric transformations resulting in a complex configuration.

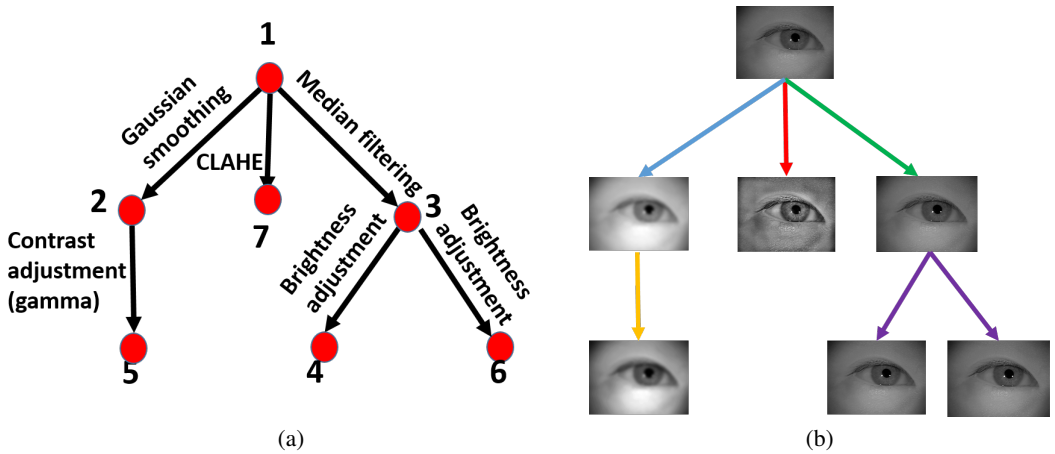


Figure 5.5: IPT based on multiple photometric transformations. (a) IPT of Breadth= 3 and Depth= 2. (b) Example of an iris image undergoing multiple transformations in a single tree (different colored lines denote the different transformations indicated in the left figure).

Table 5.2: Photometric transformations and selected range of parameters used for the second set of experiments.

Photometric Transformations	**Parameters**	**Range**
Brightness adjustment	[a, b]	$a \in [0,10]$, $b \in$ [-30, 30]
Median	[size of window]	{[3,3], [5,5], [7,7]}
CLAHE	contrast limit, Distribution	contrast limit $\rightarrow$ {0.01, 0.03}, Distribution $\rightarrow$ {'*Uniform*', '*Rayleigh*'}
Gaussian smoothing	standard deviation	*stddev* [2, 8]
Gamma correction	gamma	*gamma* [0.1, 2]

Table 5.3: Experiment 1. Performance of the 3 parametric models in representing each of the 5 photometric transformations.

	Models	Best Fit Rate (%) (Forward transformation)	Best Fit Rate (%) (Reverse transformation)	Mean PE (Forward transformation)	Mean PE (Reverse transformation)
Brightness adjustment	GL	0	0	5.1289×10^{-12}	1.854×10^{-12}
	GQ	**100**	**99.67**	1.9129×10^{-14}	2.0308×10^{-14}
	LL	0	0.33	1.4292	1.2018
Median	**GL**	0	**100**	0.0306	0.0259
	GQ	**100**	0	0.0244	0.0303
	LL	0	0	1.5227×10^{6}	3.2743×10^{7}
CLAHE	GL	0	0	0.2232	0.1344
	GQ	10.67	13.33	0.1417	0.0844
	LL	**89.33**	**86.67**	0.0864	0.0461
Gaussian smoothing	**GL**	0	**100**	0.0302	0.0229
	GQ	**87.67**	0	0.0214	0.0306
	LL	12.33	0	0.0501	0.1301
Gamma correction	GL	1.33	1.33	0.0087	0.0103
	GQ	**91.33**	**73.67**	0.0015	0.0032
	LL	7.33	25	0.1724	0.1738

5.3.2 Results and Discussion

Analysis of results in terms of parameter estimation and reconstruction accuracy is discussed in this section.

For the first set of experiments, the results are presented in Table 5.3. Here, for each photometric transformation, we report the percentage of times each of the 3 models gave the best fit. The results indicate that the GQ model best fits the Brightness and Gamma correction operations for both forward and reverse transformations. However, it should be noted that for the Brightness adjustment operation, GL model performs almost at par with the GQ model as evident from the average photometric error value reported in the last two columns. For a given image pair, we do not assume prior information about which is the original image and which one is the transformed image. As such, a model may fit well the transformation in both directions. Thus, the performance of a model (and, therefore, its relevance) can be claimed to be acceptable *only if* it results in low PE in both forward and reverse directions. As anticipated, the LL model is able to better characterize the CLAHE operation which is a local transformation.

For the second set of experiments, the IPT-DAG reconstruction accuracy is reported in Table 5.4.

The results reflect *how well a model discriminates* between the original and the transformed image. For example, the LL model which *best* fits the CLAHE transformation (see Table 5.3), results in the highest reconstruction accuracy for CLAHE as indicated in the third row of Table 5.4. The reconstruction accuracy for Brightness adjustment is identical for the GL and GQ models as the magnitude of the estimated parameters were similar (for GQ model the coefficient of the quadratic term was $\approx 10^{-14}$). The poor reconstruction accuracy for the global Gamma correction can be attributed to the failure of all the three models in representing the transformation. The contrast adjustment uses the gamma value in the range [0.1, 2] to decide the shape of the curve governing the relationship between the input and the output pixel values; thus, gamma adjustment is a non-linear mapping which cannot be aptly represented using a simple quadratic model. Another example of failure of the proposed IPT-DAG reconstruction is demonstrated for Gaussian smoothing in Figure 5.6. Such a case arises when the value of the standard deviation value used for the smoothing operation is small ($\sigma = 2.55$), and the dissimilarity measure cannot successfully discriminate between the source and the transformed image, resulting in poor IPT reconstruction. In Figure 5.6, the second image was misclassified as the source image.

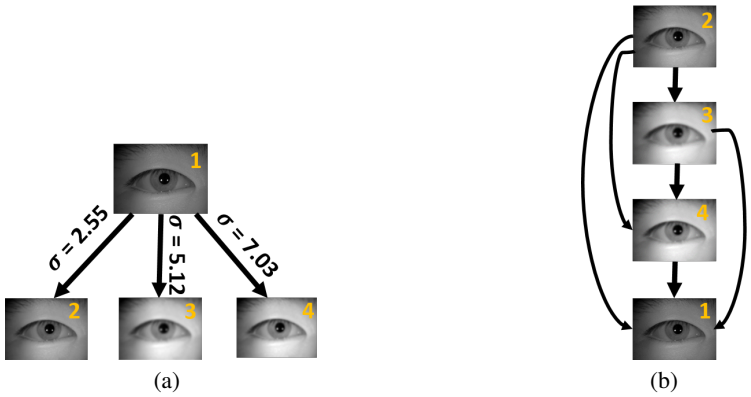


Figure 5.6: Example of an IPT of breadth 3 and depth 1 undergoing Gaussian smoothing resulting in an incorrect IPT-DAG reconstruction. (a) Original IPT-DAG (σ denotes the standard deviation governing the smoothing operation) and (b) incorrect IPT-DAG reconstruction.

Table 5.4: Experiment 2. IPT-DAG reconstruction accuracy for different tree configurations using magnitude of predicted parameters as asymmetric dissimilarity measure.

	IPT-DAG Reconstruction Accuracy (%)			
	Models	B1D3	B2D2	B3D1
Brightness adjustment	**GL**	**91.16**	**86.76**	**88.26**
	GQ	**91.16**	**86.76**	**88.26**
	LL	88.03	86.53	88.18
Median	GL	0	0	0
	GQ	**97.70**	**91.44**	**92.02**
	LL	0.43	0.93	1.33
CLAHE	GL	89.47	62	47.94
	GQ	15.36	36.17	47.11
	LL	**99.52**	**99.16**	**98.31**
Gaussian smoothing	GL	0	0	0
	GQ	**67.07**	**58.45**	**52.35**
	LL	3.25	1.92	0.98
Gamma correction	GL	22.45	17.94	14.68
	GQ	**39.58**	**40.58**	**38.95**
	LL	28.84	26.49	25.86

Table 5.5: Experiment 3. IPT-DAG Reconstruction Accuracy for the multiple transformation scenario depicted in Figure 5.5.

Models	IPT-DAG Reconstruction Accuracy (%)
GL	47.93
GQ	**71.30**
LL	46.67

5.4 Summary

In this chapter, we introduced the content-based analysis of biometric images. We construct an image phylogeny tree (IPT) that captures the relationship between a set of photometrically modified iris images. the IPT contains the original image as the root and a set of directed immediate and ancestral links depicting the order in which the images have been modified. The proposed approach used three parametric functions, namely, global linear, global quadratic and local linear for modeling photometric transformations, out of which the global quadratic model outperformed the linear models. However, the quadratic model struggled to perform to model highly non-linear transformations (gamma correction and Gaussian smoothing). This work gave us insight into

moving towards a probabilistic framework that can sufficiently discriminate between original and transformed images for accurate IPT reconstruction.

CHAPTER 6

A PROBABILISTIC FRAMEWORK FOR IMAGE PHYLOGENY USING BASIS FUNCTIONS

Portions of this chapter appeared in the following publications:

S. Banerjee and A. Ross, "Face Phylogeny Tree: Deducing Relationships Between Near-Duplicate Face Images Using Legendre Polynomials and Radial Basis Functions," 10th IEEE International Conference on Biometrics: Theory, Applications and Systems, (Tampa, USA), September 2019.

S. Banerjee and A. Ross, "Face Phylogeny Tree Using Basis Functions", IEEE Transactions on Biometrics (T-BIOM) 2020.

6.1 Introduction

In the previous chapter, we presented a deterministic approach for modeling a set of photometrically modified near-duplicate iris images. We have further realized that the application of photmetric transformations to face images portray realistic scenarios. Therefore, in this chapter, we tackle the challenging problem of image phylogeny in the context of different biometric modalities (face, fingerprint and iris images) subjected to different types of photometric transformations using a probabilistic framework. We pose the problem of asymmetric measure computation as the "likelihood ratio" problem and employ a topological sorting technique to construct the image phylogeny tree.. We further evaluate the proposed approach on a host of photometric, geometric transformations and deep learning-based transformations. The contributions of the proposed method are as follows:

1. Considering three different families of basis functions for modeling photometric and geometric transformations: (i) Orthogonal polynomial family (Legendre and Chebyshev), (ii) Wavelet family (Gabor), and (iii) Radial Basis family (Gaussian and Bump).

2. Performing cross-modality testing, *i.e.*earning the parameters of the basis functions using *face* images, and testing it on near-infrared *iris* images and optical sensor *fingerprint* images.

3. Testing on multiple IPT configurations to evaluate the robustness of the proposed method. Also, robustness to unseen photometric and geometric transformations (*i.e.,* transformations not used during the training phase) accomplished using deep learning-based schemes, as well as open-source and commercial software, is assessed. Furthermore, we have performed qualitative assessment of the IPTs reconstructed using the proposed method on near-duplicates downloaded from the internet.

4. Visualizing the results using t-distributed stochastic neighbor embedding (t-SNE) to better understand the ability of the basis functions in modeling the transformations and discriminating between forward and reverse transformation directions.

5. Employing von Neumann directed graph entropy to better understand and evaluate the reconstructed IPTs.

6.2 Proposed Method

A photometrically related image pair $(\boldsymbol{I}_i, \boldsymbol{I}_j)$ can be generated by applying a single transformation or a sequence of transformations to one image resulting in the other image. However, to construct the IPT we require to differentiate between the original image and the transformed image. Say, if $\boldsymbol{I}_i$ is the original image and $\boldsymbol{I}_j$ is the transformed image, then the IPT should have a directed link as follows: $\boldsymbol{I}_i \rightarrow \boldsymbol{I}_j$. Applying this same principle to a set of near-duplicate photometrically related images, we need two sets: the first set denoting the *parent* nodes and the second set denoting the *child* nodes. These two sets are then used to construct the IPT ($parent \rightarrow child$). So the first step is to identify the sets of parent and child nodes from an array of near-duplicates.

We proceed to identify the parent and the child nodes for each pair of images by first modeling the transformation that relates the two images. We use parameterized basis functions to model the transformations in both directions ($\boldsymbol{I}_i \rightarrow \boldsymbol{I}_j$ and $\boldsymbol{I}_i \leftarrow \boldsymbol{I}_j$). But modeling the transformation does not indicate which is the parent node and which is the child node. To accomplish this, we require an asymmetric measure to distinguish between the forward and reverse directions. We

pose the asymmetric measure computation as the *likelihood* ratio problem [20]. To compute the likelihood ratio, we adopt a supervised framework with a training phase and a testing phase. In the *training* phase, we model numerous transformations for a large number of near-duplicate image pairs in both directions (in this phase we know the original and the transformed images *apriori*). This results in two sets of parameter distributions, one for the forward transformation and the other for the reverse transformation. In the *testing* phase, for a given near-duplicate pair, we first model the transformations in both directions. Next, we use the estimated parameters to determine how *likely* they are to originate from the forward parameter distribution as opposed to the reverse parameter distribution. This step leads to the computation of the asymmetric similarity measure. We repeat this step for all image pairs in the near-duplicate set. Upon pairwise modeling of all the near-duplicate images in the set, we perform thresholding to identify *related* image pairs. The similarity measure is then utilized to identify which image from the related pair is the parent, thus, making the other image its respective child. The sets of parent and child nodes are ultimately used to generate the IPT.

In this work, we seek to model an arbitrary transformation using a set of parametric functions, that we refer to as basis functions. Such an approach is needed since the space of photometric transformations is very vast; further, each of these transformations has a large number of parameter values. For example, a simple brightness adjustment can be accomplished using a large number of brightness values. The use of a fixed set of parametric functions to approximate a transformation reduces the otherwise complex task of modeling the photometric transformation. Thus, the task of approximating the transformations involves learning the *parameters* of the basis functions, subject to a criterion. In our case, the criterion or the objective function is the minimization of the photometric error between a near-duplicate image pair. This is formulated as below:

$$PE(\boldsymbol{I_i}, \boldsymbol{I_j}) = \min_{\boldsymbol{\alpha}} \Sigma_p \|\boldsymbol{I}_i(p) - \mathcal{T}[\boldsymbol{I}_j(p)|\boldsymbol{\alpha}]\|_2^2. \tag{6.2.1}$$

Here, $\mathcal{T}[\cdot|\boldsymbol{\alpha}]$ denotes the photometric transformation. We model the transformation using the basis function as $\boldsymbol{I}_i(p) \approx \mathcal{T}[\boldsymbol{I}_j(p)|\boldsymbol{\alpha}] \approx \sum_{h=1}^{m} \alpha_h \mathbb{B}_h[\boldsymbol{I}_j(p)]$, where the transformation is applied to each pixel p. $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$ is the parameter vector to be estimated and m is the number

of basis functions. In this work, we have five different types of basis functions, so the value of m depends upon the choice of the basis function. Next, we describe the process of modeling the transformations using the basis functions and the parameter estimation routines.

6.2.1 Parameter Estimation of Basis Functions

6.2.1.1 Orthogonal Polynomial Basis Functions

1. **Legendre polynomials** are a class of orthogonal polynomials defined in the interval [-1, 1]. The Legendre polynomial of degree n computed at x is denoted as $P_n(x)$ and is written as follows:

$$L_n(x) = 2^n \sum_{k=0}^{n} x^k \binom{n}{k} \binom{\frac{n+k-1}{2}}{n}. \tag{6.2.2}$$

   Legendre polynomials have been successfully used for image template matching [130], and image reconstruction and compression [104]. Note that Eqn.(6.2.2) simplifies to a linear function for $n = 1$ and a quadratic polynomial for $n = 2$.

2. **Chebyshev polynomials** are a special case of Jacobi polynomials defined in the interval [-1, 1]. There are two kinds of Chebyshev polynomials, here we are interested in Chebyshev polynomials of first kind which have been extensively used for approximating complex functions such as graph convolution [55]. The explicit representation is presented below:

$$C_n(x) = x^n \sum_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \binom{n}{2k} (1 - x^{-2})^k. \tag{6.2.3}$$

In the notation, $\sum_{h=1}^{m} \alpha_h \mathbb{B}_h[\boldsymbol{I}_j(p)]$, $\mathbb{B}_h[\cdot]$ equals $L_h(\cdot)$ if Legendre polynomial is used and $C_h(\cdot)$ if Chebyshev polynomial is employed, and $m = n + 1$. Next, we solve the objective function in Eqn.(6.2.1) using the inverse compositional estimation (ICE) algorithm [16, 26]. The IC update rule expresses the updated transformation as a composition of the current transformation and the inverse of the incremental transformation. See [26] for a detailed derivation of the IC update rule. The parameter $\boldsymbol{\alpha}$ is computed as $\boldsymbol{\alpha} \leftarrow \frac{\boldsymbol{\alpha}_{old}}{1 + \Delta\boldsymbol{\alpha}}$. The IC update rule is an iterative optimization

algorithm and updates the new $\boldsymbol{\alpha}$ using the previous value, $\boldsymbol{\alpha}_{old}$, and the incremental $\Delta\boldsymbol{\alpha}$. The incremental parameter vector is computed as,

$$\Delta\boldsymbol{\alpha} = (\boldsymbol{J_S}\boldsymbol{J_S}^T + \lambda\boldsymbol{Id})^{-1}\boldsymbol{J_S}\boldsymbol{E}. \tag{6.2.4}$$

Here, the term $\boldsymbol{J_S}$ is known as the Jacobian of the source image ($\boldsymbol{I}_i$), and the term ($\boldsymbol{J_S}\boldsymbol{J_S}^T$) is known as the approximate Hessian matrix. We applied L_2 regularization to the Hessian matrix. Here, λ denotes the regularization parameter, $\boldsymbol{Id}$ denotes the identity matrix, and $\boldsymbol{E}$ denotes the error image computed between the source image ($\boldsymbol{I}_i$) and the modeled target image ($\mathcal{T}[\boldsymbol{I}_j|\boldsymbol{\alpha_{old}}]$). In this work, $\boldsymbol{\alpha}$ is a 6-dimensional vector for both Legendre and Chebyshev polynomials.

6.2.1.2 Wavelet Basis Functions

Gabor wavelets are used for extracting texture information from images [49] and has been selected as one of the basis functions for modeling the transformations. We employed a bank of Gabor filters parameterized with the wavelength and orientation. A set of four discrete wavelengths $\{2, 3, 4, 5\}$ and four orientations $\{0°, 45°, 90°, 135°\}$ are selected. Each wavelength corresponds to a single filter scale that treats the image at a different resolution. Thus, we have a bank of sixteen Gabor filters. We filtered the image with the Gabor bank and we obtained 16 filtered responses. However, in our case we combined the orientation responses for each wavelength, thus reducing the total number of responses from 16 to 4. Finally, we use ICE to estimate the 4-dimensional parameter vector $\boldsymbol{\alpha}$ ($m = 4$).

6.2.1.3 Radial Basis Functions

The polynomial and wavelet basis functions model the transformations at pixel level. In pixel-level modeling, the photometric error between each pixel of the original and transformed image pair is minimized using a weighted linear combination of basis functions. However, local filtering operations such as median filtering are applied in a patch-wise manner. Therefore, we used the family of radial basis functions that possesses nice smoothing properties to model transformations

at the patch level. In patch-level modeling, the photometric error between two patches, one patch belonging to the original image and the second patch belonging to the transformed image, is minimized using a weighted linear combination of basis functions. Therefore, patch-level modeling considers all the pixels within a patch for minimizing the photometric error. This resolves the spatial dependencies observed in patch-based photometric transformations.

1. **Gaussian radial** kernel is the first type of smoothing functions considered in the work. Gaussian RBF computed at x is denoted as $K(x)$ and is written as $K(x) = \exp \|x - \mu\|^2$. Consider an image pair $(\boldsymbol{I}_i, \boldsymbol{I}_j)$ each of which is tessellated into N_P non-overlapping blocks of size 16×16. For the block q, where $q = [1, \cdots, N_P]$, let $\boldsymbol{I}_j^q(p) = \mathcal{T}[\boldsymbol{I}_i^q(p)|\boldsymbol{\alpha}_q] \approx \sum_p \alpha_{p,q} K[\boldsymbol{I}_i^q(p)]$. Here, p denotes the pixel intensity value within the q^{th} block and μ indicates the average of the pixel intensity values within that block. For Gaussian RBF, $m = p$, *i.e.*or a 16×16 block, the local least squares estimation yields $\boldsymbol{\alpha}_q$. Simplifying using the matrix notation yields $\boldsymbol{I}_j^q \approx \boldsymbol{\alpha}_q^T \mathbb{B}[\boldsymbol{I}_i^q]$, where $\mathbb{B}[\boldsymbol{I}_i^q] = K[\boldsymbol{I}_i^q(p)], \forall p$. The local least squares method is used to estimate the coefficient vector $\boldsymbol{\alpha}_q$ for each block. The final $\boldsymbol{\alpha}$ is a 256-dimensional vector obtained by computing the average of all $\boldsymbol{\alpha}_q$s. See [20] for detailed derivation.

2. **Bump RBF** is a smooth compact function which can be interpreted as a Gaussian function scaled to lie on a disc of unity radius. It is not analytic unlike Gaussian RBFs, but can be used as generalized functions which are essential in converting discontinuous functions to smooth functions [153]. In this case, $K(x) = \exp\left(-\frac{1}{1-x^2}\right)$ for $x \in [-1, 1]$. Here, x is mean-centered. Using least squares estimation, we obtain $\boldsymbol{\alpha}$ (256-dimensional).

6.2.2 Asymmetric Measure Computation and IPT Construction

The asymmetric measure can be in the form of pairwise similarity or dissimilarity, but in this work, we adopt a similarity-based asymmetric measure. The similarity measure computed between a pair of images determines whether an image pair is photometrically related or not, *i.e.*hether a link

should exist between a pair of nodes (images) in the IPT; secondly, determine the direction of the link by identifying the parent node and the child node. The parameters estimated for modeling the transformations are utilized to compute this similarity measure as described below.

6.2.2.1 Likelihood ratio for computing the asymmetric similarity measure

Given a pair of images, $(\boldsymbol{I}_i, \boldsymbol{I}_j)$, we first estimate the parameter vectors $\boldsymbol{\alpha}_{ij}$ and $\boldsymbol{\alpha}_{ji}$ in both directions ($\boldsymbol{I}_i \rightarrow \boldsymbol{I}_j$ and $\boldsymbol{I}_j \rightarrow \boldsymbol{I}_i$). The parameter vectors are necessary but not sufficient for constructing the IPT. We compute the *likelihood ratio* from the estimated parameters to yield a similarity score which can discriminate between the forward and reverse directions, and can thus be used to construct the IPT. To compute the likelihood ratio, we need the probability distribution of the parameter vectors — $p_f(\boldsymbol{\alpha})$ and $p_b(\boldsymbol{\alpha})$ corresponding to forward and reverse directions for a large number of training images. The probability distributions are generated in a supervised fashion, where we assume that we know for an image pair from the training set $(\boldsymbol{I}_r, \boldsymbol{I}_s)$, $\boldsymbol{I}_r$ is the original image and $\boldsymbol{I}_s$ is the transformed image. Then the forward transformation refers to $(\boldsymbol{I}_r \rightarrow \boldsymbol{I}_s)$, and the reverse transformation refers to $(\boldsymbol{I}_s \rightarrow \boldsymbol{I}_r)$. The set of $\boldsymbol{\alpha}_{rs}$ vectors computed for a large number of image pairs are used to estimate $p_f(\boldsymbol{\alpha})$. Similarly, the set of $\boldsymbol{\alpha}_{sr}$ parameter vectors are used to determine $p_b(\boldsymbol{\alpha})$. We utilized Parzen window based non-parametric density estimation scheme [140] to obtain $p_f(\boldsymbol{\alpha})$ and $p_b(\boldsymbol{\alpha})$.

Upon obtaining the forward and the reverse parameter distributions, we now compute the likelihood ratios as follows: $\Lambda_{ij} = \dfrac{p_f(\boldsymbol{\alpha_{ij}})}{p_b(\boldsymbol{\alpha_{ij}})}$. Similarly, $\Lambda_{ji} = \dfrac{p_f(\boldsymbol{\alpha_{ji}})}{p_b(\boldsymbol{\alpha_{ji}})}$. Our intuition is that we will observe a higher value of Λ_{ij} compared to Λ_{ji}, if $\boldsymbol{I}_i$ is the original image and $\boldsymbol{I}_j$ is the transformed image. In this case, $\boldsymbol{\alpha_{ij}}$ belongs to the forward distribution and should result in a higher value of Λ_{ij}. Conversely, $\boldsymbol{\alpha_{ji}}$ belongs to reverse distribution, resulting in a lower value of Λ_{ij}. The likelihood ratios are further used to populate the similarity matrix.

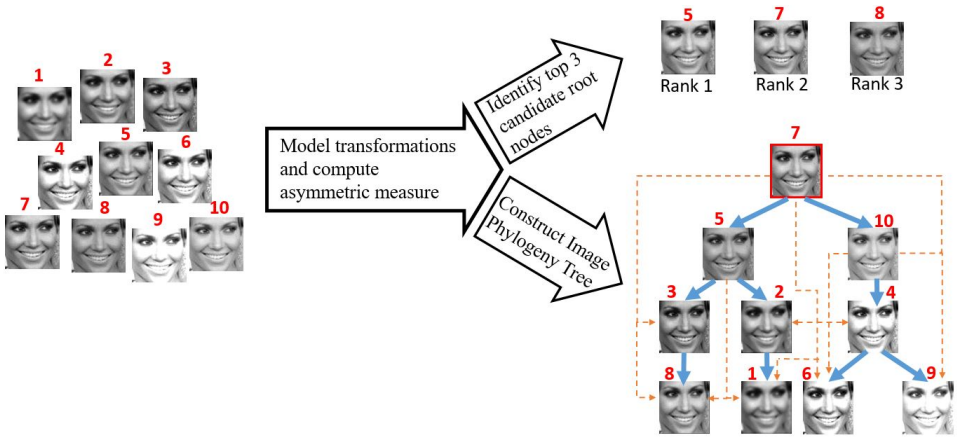


Figure 6.1: The outline of the proposed method. The proposed method first models the photometric transformations between every image pair and then computes the asymmetric measure. Given a set of near-duplicate images as input (on the left) the two objectives are: (i) to determine the candidate set of root nodes, and (ii) to construct the IPT when the root image is known. The dashed arrows indicate ancestral links and the bold arrows indicate immediate links between parent and child nodes.

6.2.2.2 IPT Construction

The similarity matrix $\boldsymbol{S}$ of size $n \times n$ is populated by the likelihood ratio values as follows: $\boldsymbol{S}_{ij} = \Lambda_{ij}; i, j = [1, \cdots, n]$ and $i \neq j$. The diagonal elements of the similarity matrix are ignored as we do not consider self-loops in an IPT. The similarity matrix is then employed for (i) identifying the candidate root nodes and (ii) constructing the IPT. The steps are described below.

1. **Indicator matrix representation:** This step helps in pruning the outliers which can be falsely identified as root nodes. The indicator matrix is constructed by thresholding the similarity matrix against a suitable threshold, which results in a binary matrix. The details of the threshold selection are described in [20]. The indicator matrix serves as an adjacency matrix of a coarse directed acyclic graph that is further refined for constructing the IPT. The reason it is referred to as coarse is that it may contain some spurious edges.

2. **Candidate root nodes identification:** In this work, we consider each IPT to have a single root node. The authors in [57, 62] considered each node as a potential root, one at a time, and then computed a cost function for each IPT constructed using the potential root. The

IPT resulting in the least cost function value was selected, and its root node was used for the final evaluation. This involves $O(n^3)$ (can be optimized to $O(n^2)$) complexity as reported in [57]. In contrast, the method proposed here computes a set of three candidate root nodes, which corresponds to the top 3 choices for roots out of n nodes. This requires finding the nodes having the highest number of 1's in the indicator matrix (we consider ancestral edges as correct edges). The entire process requires summing each row of the indicator matrix followed by sorting and this results in $O(n \log n)$ computational complexity.

3. **IPT generation:** We construct the IPT as described in [20] using a depth-first search-based tree spanning technique. The choice of depth-first search (DFS) over breadth-first search (BFS) is motivated by the fact that DFS has a linear memory requirement with respect to the nodes and results in a faster search ($O(n)$) and is therefore used for topological sorting. The total computational complexity for the IPT construction using the proposed method is $O(n \log n) + O(n) \approx O(n \log n)$.

The outline of the proposed method for constructing the IPT is illustrated in Figure 6.1.

Table 6.1: Description of the datasets used in this work.

Modality	Name of the Dataset	Dataset Identifier	No. of subjects	No. of images
Face	LFW	Partial Set	391	12,290
		Full set	468	27,270
Iris	CASIA-IrisV2 Device2	—	37	7,260
	CASIA-IrisV4 Thousand	—	525	5,005
Fingerprint	FVC 2000 DB3	Config I	110	8,800
		Config II	90	7,200

6.3 Experiments

In this section, we describe the datasets employed, the experiments conducted and finally report the results.

Table 6.2: Photometric transformations and the range of the corresponding parameters used in the training and testing experiments. The transformed images are scaled to $[0, 255]$. Note that experiments were also conducted using other complex photometric transformations besides the ones listed here.

Photometric Transformations	**Level of Operation**	**Parameters**	**Range**
Brightness adjustment	Global	[a,b]	a ∈ [0.9,1.5], b ∈ [-30,30]
Median filtering	Local	size of window [m,n]	m ∈ [2,6], n ∈ [2,6]
Gaussian smoothing	Global	standard deviation	stddev ∈[1,3]
Gamma transformation	Global	gamma	gamma ∈ [0.5,1.5]

6.3.1 Datasets

We have used four datasets belonging to three different modalities to conduct experiments. For the face modality, we used images from the Labeled Faces in the Wild (LFW) dataset [88]. For the iris modality, we used near-infrared iris images from the CASIA-IrisV2 Device2 subset [6] and CASIA-IrisV4 Thousand subset [7]. For the fingerprint modality, we used images from the FVC2000 DB3 dataset [114]. The description of the datasets is provided in Table 6.1. We have selected four photometric transformations, *viz.*, Brightness adjustment, Median filtering filtering, Gaussian smoothing, and Gamma transformation as used in [20] to test the proposed IPT construction algorithm. The parameter range for each of the transformations is described in Table 6.2.

6.3.2 Experimental Methodology

We have performed seven experiments which are described below.

6.3.2.1 Experiment 1: Efficacy of basis functions

In this experiment, we evaluate the ability of the basis functions to (i) model the photometric transformations and (ii) discriminate between the forward and reverse directions. To accomplish the first task, we perform two evaluation methods. The first evaluation involves *deterministically* selected parameters, whereas the second evaluation involves *randomly* selected transformation parameters. For the first evaluation, we select a face image, $\boldsymbol{I}$ and subject it to a single transformation, *e.g.*, gamma adjustment, parameterized with a specific γ value, resulting in $\boldsymbol{I'}$. We repeat this

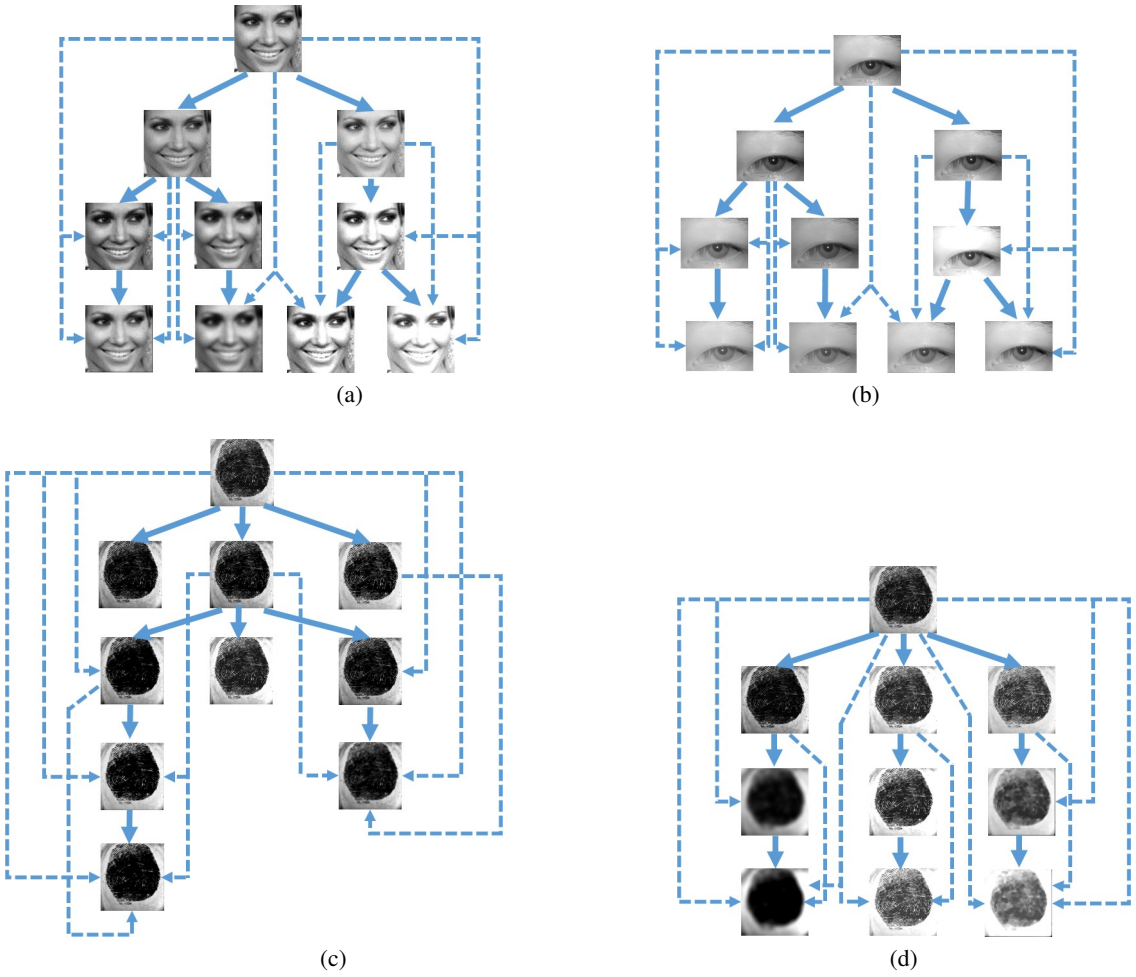


Figure 6.2: IPT configurations used in Experiments 2 and 3 for the face, iris and fingerprint modalities. Note that the same configuration was tested across two modalities (Face and Iris) while, two different configurations were tested for the same modality (Finger). The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.

process 200 times, each time we use an incrementally modified γ_{new} ($\gamma_{new} = \gamma_{old} + \Delta\gamma$), thus, resulting in 200 near-duplicate image pairs. Furthermore, we repeat this process for 5 images corresponding to 5 different subjects. Therefore, we have a total of 1,000 photometrically related image pairs for a single transformation. We conduct this process for each of the four transformations indicated in Table 6.2. Next, we use the basis functions to model the transformation *only* in the forward direction in this experiment. Then, we use t-SNE [156] to reduce the dimensionality of the estimated vectors and project them onto 3-dimensions. This experiment is conducted to assess the ability of the basis functions in modeling the transformations and we visually interpret

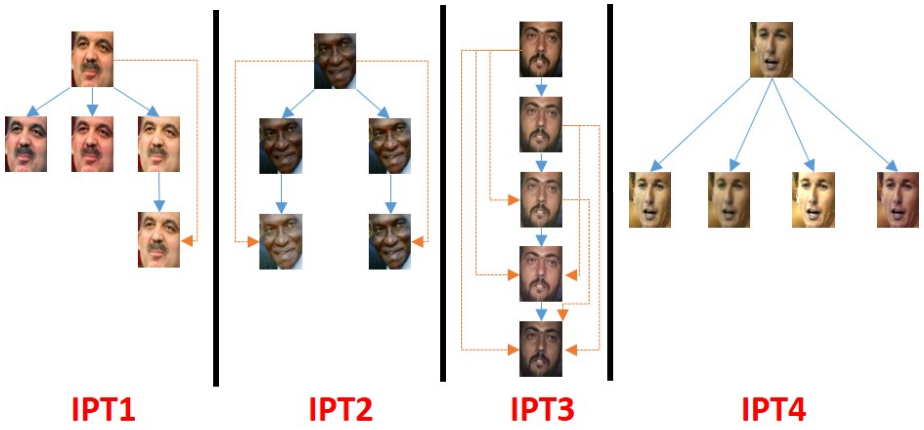


Figure 6.3: IPT configurations used in Experiment 4. The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.

the results. The second protocol involves modeling 2,000 image pairs in the forward direction using the five basis functions, where each of the 500 images were subjected to each of the four photometric transformations using randomly selected parameters. We further computed the residual photometric error (PE), for each image pair which is the difference in the pixel intensity between the actual target image and the output modeled using the basis functions. We then computed the mean over all 2,000 image pairs. The mean of the residual PE evaluates the ability of the basis functions to accurately model the transformations.

To evaluate the second task, *i.e.*iscriminating between the forward and reverse directions, we selected 400 image pairs (100 image pairs corresponding to each of the four transformations) from the 2,000 pairs, that were generated using randomly selected parameters. Then, we modeled the transformations in *both* forward and reverse directions, and estimated the parameters. We further used t-SNE to obtain a 2-dimensional embedding of the estimated parameters, and visualized the projections to analyze the performance of the basis functions.

6.3.2.2 Experiment 2: IPT Reconstruction

In this experiment, we evaluated the proposed approach in terms of (i) Root identification and (ii) IPT Reconstruction accuracy metrics as used in [20]. We followed the same experimental protocol in [20], and assessed the performance of the basis functions on both a partial set and a full-set of face images from the LFW dataset.[1] We used the IPT configuration presented in Figure 6.2(a) for this experiment.

6.3.2.3 Experiment 3: Cross-modality testing on multiple configurations

We tested the proposed approach on iris images and fingerprint images. This experiment is intended to demonstrate the generalizability of the proposed method across modalities.

1. **Iris Images** —We applied a random sequence of photometric transformations on near-infrared iris images. We evaluated the root identification and IPT reconstruction accuracies for 726 IPTs using the same procedure as described in Section 6.2.2. We used the same IPT configuration as the one used for face images (see Figure 6.2(b)). Note, the parameter probability distributions in the forward and reverse directions are computed using a training set comprising of *face* images; the test images are *iris* images. The test iris images are acquired in the near-infrared spectrum, in contrast to the training face images that are acquired in the visible spectrum. As a result, this experiment can also be treated as an assessment of the basis functions for cross-spectral modeling.

2. **Fingerprint Images** —Two different IPT configurations are tested as depicted in Figures 6.2(c) and 6.2(d). This experiment tests the generalizability of the proposed approach as a function of the breadth and depth of the IPT. We refer to Figure 6.2(c) as Config I and Figure 6.2(d) as Config II. The IPT configuration used for testing the face and iris images is more balanced (similar distribution of nodes on the left and the right sides of the root)

[1] http://iprobe.cse.msu.edu/dataset_detail.php?id=1&?title=Near-Duplicate_Face_Images_(NDFI)

compared to the Config I structure which has more depth than breadth, whereas Config II has the same breadth at successive depths.

We also performed an intra-modality experiment which serves as the baseline experiment to compare against the performance of the cross-modality experiment. The training and testing partitions for the intra-modality experiments are as follows:

- Iris images — We used 5,005 images from the CASIA-IrisV4 Thousand subset belonging to 525 subjects to learn the parameter distributions in the forward and the reverse directions. We then tested it on 726 IPTs (same configuration as in Figure 6.2(b)) constructed from the CASIA-IrisV2 Device2 subset. This experiment can also be considered as a *cross-dataset* experiment, due to the use of two different datasets in the training and testing phases.

- Fingerprint images — We used the same dataset (*intra-dataset*) in training and testing but we strictly followed a subject disjoint protocol. This, however, resulted in a lesser number of training images. 560 images from 70 subjects were used for creating parameter distributions and then tested on 3,200 images from 40 subjects. We used the same configurations as depicted in Figures 6.2(c) and 6.2(d).

6.3.2.4 Experiment 4: Robustness to unseen photometric transformations

We have considered a closed set of 4 transformations in the training stage. However, a gamut of image and video editing tools such as Photoshop, GIMP and Snapchat filters exist which can be used for image manipulation, particularly for face images. In this context, we constructed a small test set of images transformed using Photoshop operations (Hue and Saturation adjustment, Curve transformation, Color balance, and Blur filters) to create 35 IPTs corresponding to 5 subjects. The IPT configurations are selected such that they cover diverse breadth and depth values possible for an IPT with 5 nodes. See Figure 6.3. The trained parameter distributions did not encounter instances of Photoshopped images; hence, this experiment will demonstrate the robustness of the basis functions in handling unseen transformations.

6.3.2.5 Experiment 5: Ability to handle geometric transformations

We designed this experiment to assess the ability of basis functions in modeling geometric transformations. We have selected some well-known geometric transformations such as sampling using linear interpolation and affine transformations that include translation, scaling and rotation. We have selected these particular transformations as they have also been utilized in [61] for creating near-duplicates. The details about the geometric transformations and their respective parameter ranges are described in Table 6.3. We randomly selected 500 images belonging to 97 subjects from the Labeled Faces in the wild (LFW) dataset [88]. We then applied four geometric modifications (see Table 6.3) on each of these images in a random sequence with random parameter values to create 500 image phylogeny trees (IPTs). Each IPT contains 10 images so we have a total of 5,000 images. An example IPT consisting of geometrically modified images is presented in Figure 6.4. Note that the IPT configuration is the same as the one used to evaluate photometrically modified images. We have conducted the experiment using the following two protocols, and evaluated the performance using root identification accuracy at Ranks 1, 2 and 3 and IPT reconstruction accuracy.

1. The first protocol involves training on the photometrically modified images, while testing on geometrically modified images. In this protocol, the training set did not include any geometrically modified images, so it assesses the robustness of the basis functions on different classes of transformations (*i.e.*, photometric versus geometric). We have *not* modified the asymmetric measure computation method or the tree-spanning method used in constructing the IPT.

2. The second protocol involves training and testing on geometrically modified images. To accomplish this task, we created a new training set of 5,865 pairs of original and geometrically transformed images using the LFW dataset. The objective is to evaluate the performance of the basis functions when trained and tested on geometric modifications, unlike in the first protocol.

We have compared the performance of the proposed method with a baseline algorithm described in [61]. The baseline algorithm uses Speeded-Up Robust Features (SURF) and RANSAC algorithm for the task of geometric registration, followed by color channel normalization and used the residual photometric error as the asymmetric measure. The Oriented Kruskal algorithm is used for spanning

Table 6.3: Experiment 5: Geometric transformations and their parameter ranges used in this work.

Geometric transformations		**Parameters**
Re-sampling		[90%, 110%]
Generic Affine	Rotation	[-5°, 5°]
	Translation	[5, 20]
	Scaling	[90%, 110%]

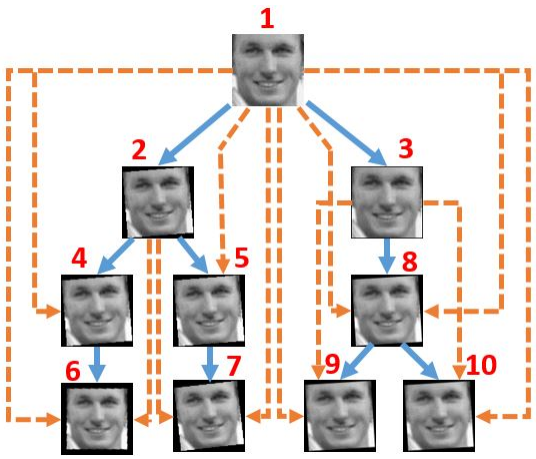


Figure 6.4: Experiment 5: An example IPT generated using geometrically modified near-duplicate images. The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.

the IPT which is a minimal spanning tree in their case. We implemented the baseline algorithm in two ways.

(a) Firstly, we used SURF and M-SAC (M-estimator sample and consensus scheme which is an improved variant of RANSAC) for geometric registration. We did not perform color channel normalization since we used gray-scale images. We then rescaled pixel intensities in the original and modified images to $[0, 255]$ prior to evaluation. We used the Oriented Kruskal algorithm for constructing the IPT.

(b) Secondly, we used the best performing basis function to compute the likelihood ratio to be used as the asymmetric measure and then employed the Oriented Kruskal for spanning the IPT.

It is important to note that, unlike the baseline method, the proposed method does not require any separate geometric registration for modeling the geometric transformations.

6.3.2.6 Experiment 6: Ability to handle near-duplicates available online

Near-duplicate images of celebrities and political figures are widely circulated on the internet. The actual sequence of generation of such near-duplicates may be unknown, but these images represent pragmatic scenarios where the ground truth may not be always available. In this experiment, we analyze how the proposed asymmetric measure and IPT construction methods can handle such images. To this end, we followed the suggestion presented in [68] and used Google image search to download 40 near-duplicates retrieved from the following 5 queries: *Angelina Jolie, Kate Winslet, Superman, Britney Spears* and *Bob Marley*. We used training parameter distributions learnt from both geometrically and photometrically modified images for each of the five basis functions used in this work. We then used the proposed asymmetric measure computation method to identify top 3 candidate root nodes. For each of the candidate root node we then reconstructed an IPT. Due to unavailability of ground truth, we could not evaluate the accuracy of the reconstructed IPTs, but we present qualitative assessment of the reconstructed IPTs.

6.3.2.7 Experiment 7: Ability to handle deep learning-based transformations and image augmentation schemes

Several deep learning-based transformations and image augmentation packages are available that can be used for applying sophisticated transformations to images in an automated fashion generating a large number of near-duplicates. In this experiment, we used a deep learning-based autoencoder [80] and open source image augmentation packages [5] used for training deep neural networks to evaluate the proposed method. We have conducted the experiment using two protocols.

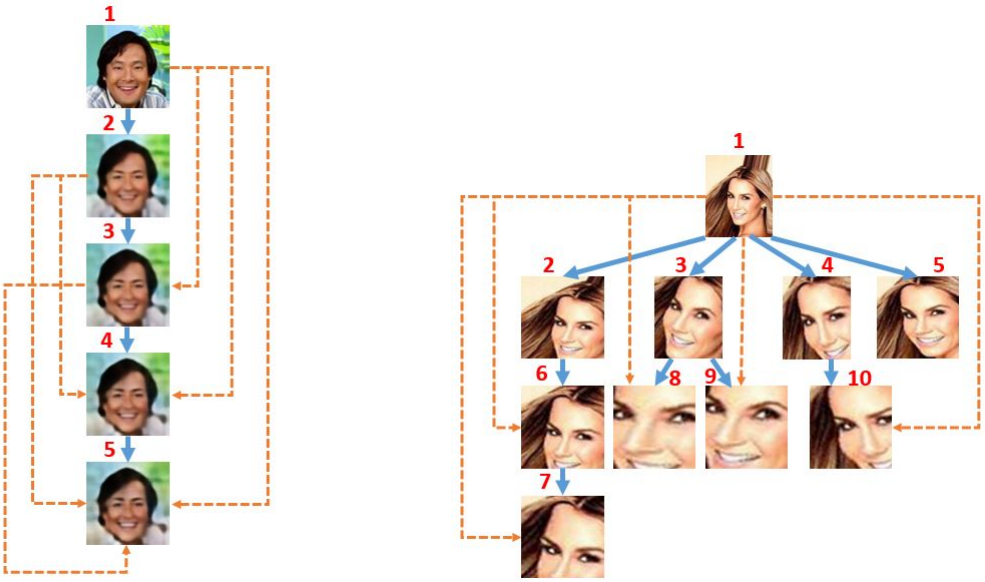
1. The first protocol involves a deep convolutional autoencoder [80]. The autoencoder was trained on $\sim 19,000$ images from the CelebA dataset [8] to generate 80 near-duplicate images belonging to 16 subjects. The resultant IPT configuration is depicted in Figure 6.5(a). The convolutional autoencoder comprises of an encoder block that consists of five convolutional layers, followed by ReLU after each convolutional layer, and the decoder block comprises of traditional

convolutional layers and nearest-neighbor based upsampling.[2] We did not use de-convolution or transposed convolution layers, as they can lead to checkerboard artifacts. The intuition behind using an autoencoder for generating near-duplicate images is to leverage upon its ability to perform high fidelity reconstruction of original input images. This fits the definition of 'near-duplicates' in our image phylogeny task, and has therefore been used in this experiment. We apply the original image as an input to the autoencoder to generate the first set of near-duplicates at depth=1. This first set of reconstructed images are again fed as input to the same autoencoder to generate near-duplicates at depth=2, and so on until we generated near-duplicates for depth=5.

2. The second protocol involves Augmentor [5], a data augmentation tool used in training deep neural networks. We used this tool, which is an open source package in Python, that applies random distortions such as zoom, cropping, rotation, re-sampling and elastic deformations to an image. See Figure 6.5(b). Some of these image transformations or the diverse parameter ranges (training involved rotation values in the interval of $[-5^\circ, 5^\circ]$, whereas, testing using Augmentor involved rotation values in the interval of $[-10^\circ, 10^\circ]$) are not encountered during the training stage. We randomly selected 100 images belonging to 100 subjects of the CelebA dataset. We applied the Augmentor on each of these 100 images to create 100 IPTs. Each IPT contains 10 images. So we tested on a total of 1,000 near-duplicate images.

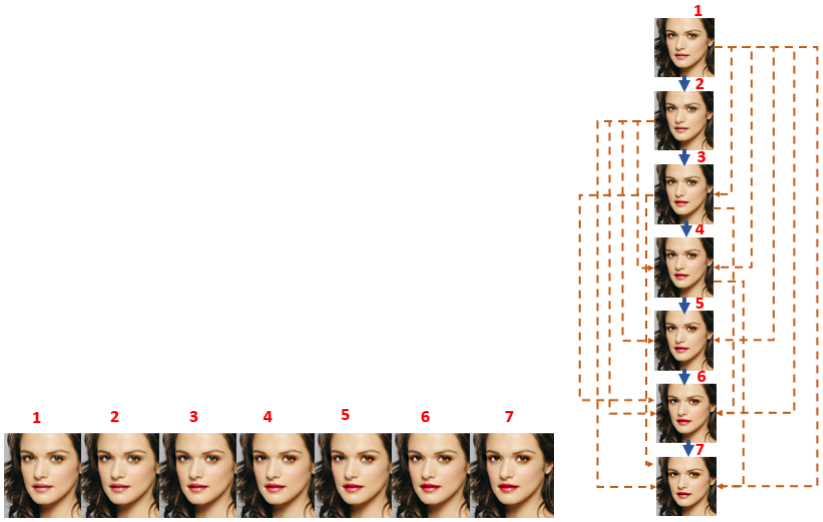
In addition, we also used some images synthesized using a deep learning-based generative network known as BeautyGlow [39]. The generative network performs a style transfer on the makeup of the individual in face images, resulting in near-duplicates as shown in Figure 6.6(a). Images are generated by sequentially increasing the magnification value of the makeup, highlighting the intensity of the makeup. We used 13 IPTs (each IPT contains 7 images), resulting in a total of 91 images.

[2] https://sebastianraschka.com/deep-learning-resources.html



(a) (b)

Figure 6.5: Experiment 7: (Left) IPT test configuration used for evaluation of the basis functions by employing autoencoder generated near-duplicates. (Right) IPT test configuration used for evaluation of the basis functions by employing open source image augmentation packages. The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.



(a) (b)

Figure 6.6: Experiment 7: (Left) Near-duplicates generated using BeautyGlow generative network. (Right) IPT constructed using Chebsyshev polynomials for the near-duplicates on the left. The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.

6.4 Results and Analysis

In this section, we first report the results observed for the experiments described in the previous section, and we further present our insights into the findings.

6.4.1 Results of Experiment 1

The 3D projected vectors obtained using t-SNE are illustrated in Figure 6.7 for each of the transformations modeled using the five basis functions. Each column denotes a photometric transformation, and each row denotes a basis function. As evident from the projections, the basis functions can model the majority of the transformations fairly well. The parameters governing each transformation are incrementally modified and, hence, their projections should ideally span a continuous trajectory. Also, we expect to observe this behavior irrespective of the transformations used or the identity of the subject. We indeed observe such a behavior for most of the cases, except for Gamma adjustment, where the polynomials and wavelet functions flounder. Note that median filtering requires integer parameter values (height and width of window). Therefore, in the t-SNE results (last column in Figure 6.7) we observe small clusters, depicting accurate modeling of discrete parameterized transformations. Out of all the basis functions, the radial basis functions seem to model the transformations the best. Figure 6.8 further substantiates that the RBFs are best at modeling transformations while Gabor wavelets perform relatively poorly. The RBFs result in the lowest mean residual photometric error, suggesting a more accurate modeling of the photometric transformations.

In terms of discriminability between the forward and reverse directions, the projections are almost indistinguishable in the two directions for the wavelet functions, but they are relatively better for polynomial functions and the RBFs, as evidenced in Figure 6.9. The polynomials have fairly well-separated projections, indicating their ability to discriminate between the forward and reverse directions. We anticipate that this ability will be reflected in the IPT reconstruction experiments.

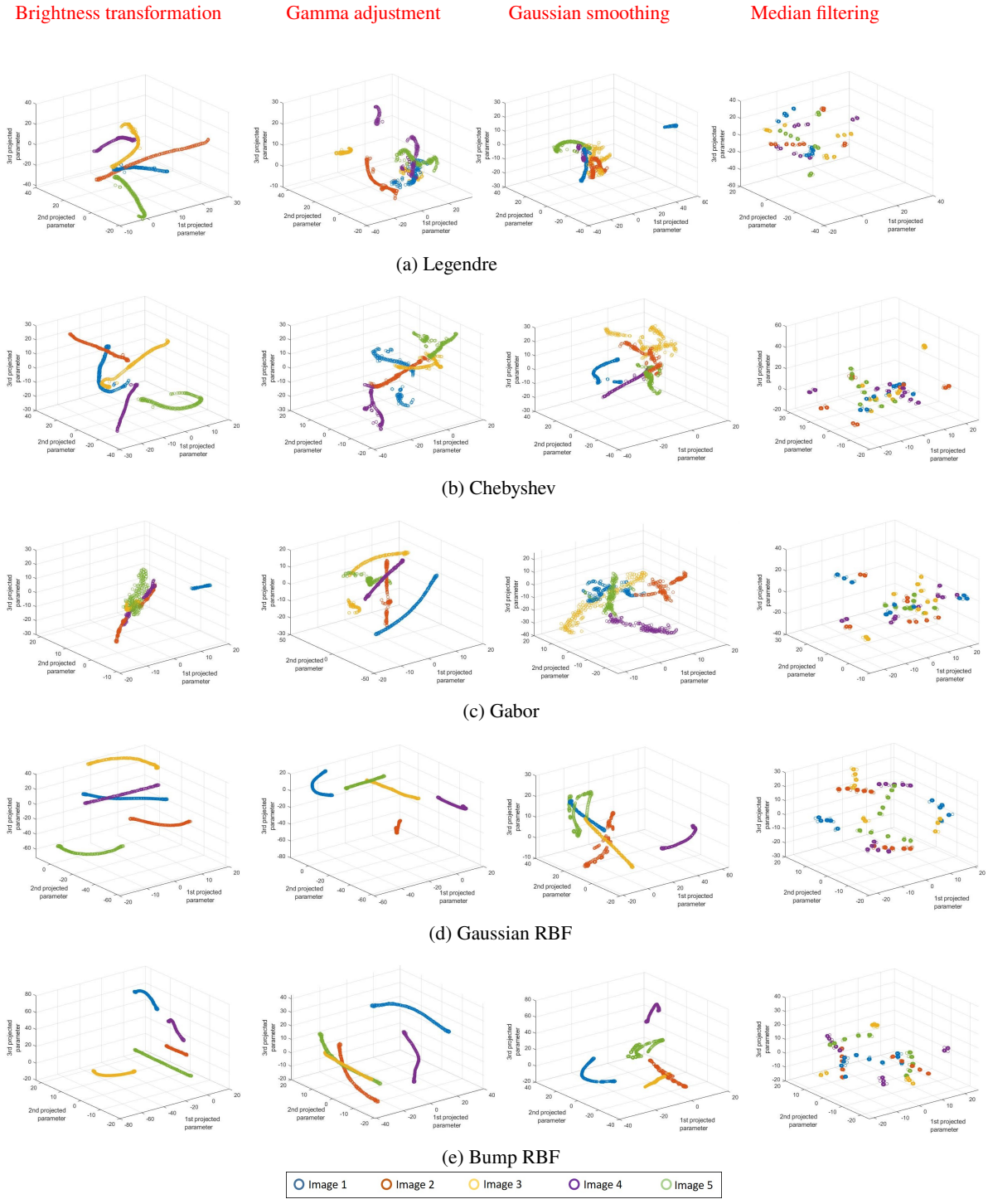


Figure 6.7: Experiment 1: 3D projected parameters using t-SNE corresponding to each photometric transformation (column) modeled using each basis function (row). Each color represents a single image. A total of 5 images were modeled. Gaussian and Bump RBFs model majority of the transformations reasonably well as indicated by the last two rows. The Brightness transformation was easiest to model as the parameters of the basis functions follow a continuous path.

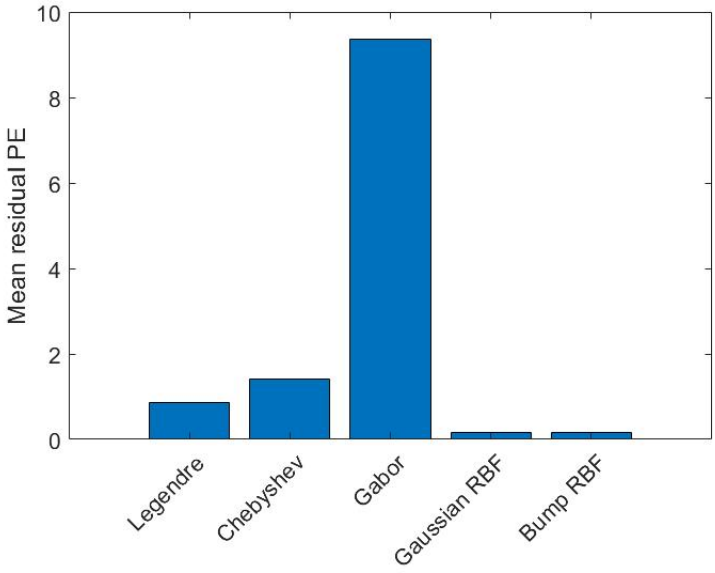


Figure 6.8: Experiment 1: The photometric error between the actual output and the the output modeled using the basis functions is denoted as residual photometric error (PE). The mean of the residual PE is demonstrated for 2,000 image pairs modeled in both forward and reverse directions using the five basis functions. Gabor resulted in the highest residual PE, and the RBFs yield the lowest residual PE demonstrating their efficacy in reliably modeling the transformations.

6.4.2 Results of Experiment 2

The results of root identification and IPT reconstruction are presented in Tables 6.4 and 6.5. Results indicate that polynomials (Legendre and Chebyshev) perform the best in a majority of cases among the set of five basis functions selected in this work. The results are consistent with the observations reported in Figure 6.9, which indicates sufficient discriminability offered by the polynomials. For the partial set, Legendre polynomials perform best both in terms of root identification (89.91%) and IPT reconstruction (70.61%) accuracies, closely followed by Chebyshev polynomials. For the full set, Gaussian RBF performs the best in terms of root identification accuracy (80.85%) while Chebsyhev polynomials perform the best in terms of IPT reconstruction accuracy (66.54%, a small improvement of $\approx$1.5% is observed compared to the results in [20]).

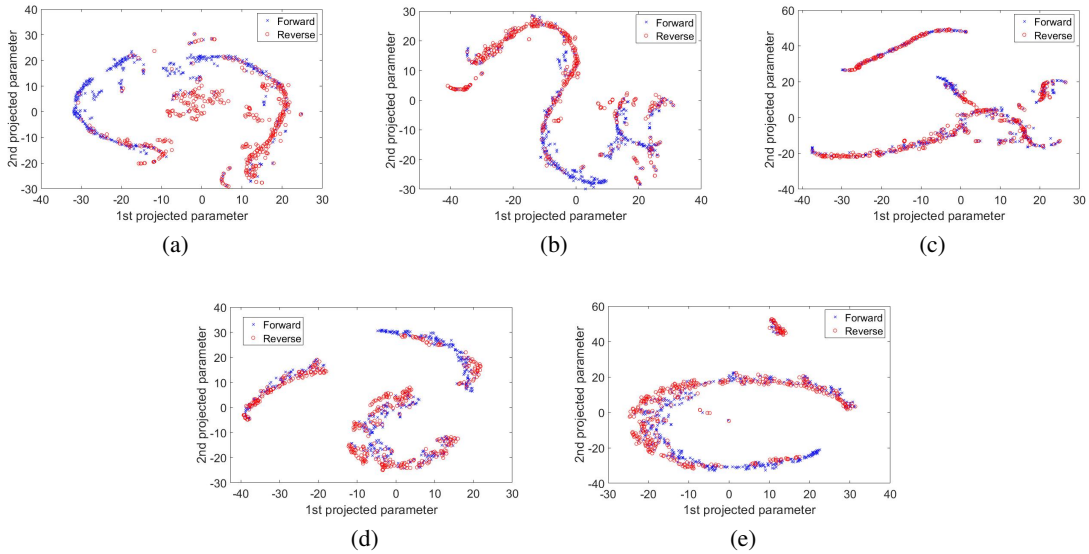


Figure 6.9: Experiment 1: 2D projected parameters using t-SNE in forward and reverse directions, corresponding to all 4 transformations modeled using each basis function: (a) Legendre, (b) Chebyshev, (c) Gabor, (d) Gaussian RBF and (e) Bump RBF. Legendre and Chebyshev polynomials can better discriminate between forward and reverse directions as indicated by the relatively well-separated parameter distributions compared to the remaining basis functions.

6.4.3 Results of Experiment 3

The results for the cross-modality experiments are presented in Table 6.6 for iris images, and in Tables 6.7 and 6.8 for fingerprint images. The purpose of this set of experiments is to assess how the proposed method performs for (i) the same IPT configuration but used across two different modalities, and (ii) the same modality but tested on different IPT configurations. Note that the training modality is different than the test modality in both cases.

The results in Table 6.6 indicate that Chebyshev polynomials obtain 94.90% root identification accuracy at Rank 3 and an IPT reconstruction accuracy of 67.90% for iris images. It is closely followed by Legendre polynomials. However, other basis functions perform poorly, specifically, the Gabor wavelets. Gabor wavelets are good texture descriptors, *i.e.*hey can extract the high-frequency features reliably from an image. In the case of photometric transformations, the pixel intensity gradient, which contributes toward high-frequency features are not significantly affected and, thus, the wavelet fails to correctly model the transformation between an image pair.

Image phylogeny on near-duplicate fingerprint images is extremely hard, as evident from the results in Tables 6.7 and 6.8. Visual inspection reveals that the set of near-duplicate fingerprint images appear to be black blobs on a white background, with no discernible differences between the set. Therefore, the root identification and IPT reconstruction accuracies are worse compared to the face and iris modalities: the best root identification performance is 65.28% and IPT reconstruction accuracy is 70.59%. This experiment also shows that the performance varies across configurations. Specifically, symmetric configurations (Config-II) can be more difficult to reconstruct than asymmetric configurations (Config-I).

Next, we compare the results of the cross-modality experiments with the baseline, which is the intra-modality experiment described in Section 6.3.2.3. The results of the baseline experiments are reported in Table 6.9. The results indicate that cross-modality experiments are commensurate with the intra-modality performance. For example, for iris images, the intra-modality experiment obtains the best root identification accuracy of 94.63% at Rank 3, while the cross-modality experiment obtains the best root identification accuracy of 94.90% at Rank 3. Furthermore, the intra-modality experiment obtains the highest IPT reconstruction accuracy of 68.62%, while the cross-modality experiment obtains the highest IPT reconstruction accuracy of 67.90% in the case of iris images.

6.4.4 Results of Experiment 4

The training set comprised of images modified using 4 rudimentary photometric transformations. In the real world, a plethora of image editing applications exists, thereby making image phylogeny for face images a typically difficult problem. We hypothesize that by creating a training dataset through random parameters on simple transformations, the unseen transformations can be reliably modeled. Results reported in Table 6.10 indicate that unseen transformations were modeled fairly well. Legendre polynomials performed the best in terms of root identification accuracy with 76.47% averaged across the four IPT configurations (see Figure 6.3). Chebyshev polynomials performed the best in terms of IPT reconstruction accuracy with 76.25% averaged across the four IPT configurations (a small improvement of $\approx$1.67% is observed compared to the results in [20]).

Table 6.4: Experiment 2: Root identification and IPT reconstruction accuracies for face images (Partial Set).

Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
Legendre	**65.90/82.18/89.91**	**70.61**
Chebyshev	53.62/74.69/85.52	70.53
Gabor	27.66/41.74/56.06	55.54
Gaussian RBF	65.25/81.04/87.79	66.15
Bump RBF	63.79/80.07/86.41	66.52

Table 6.5: Experiment 2: Root identification and IPT reconstruction accuracies for face images (Full set).

Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
Legendre	50.45/66.74/75.68	65.05
Chebyshev	45.18/65.13/76.86	**66.54**
Gabor	29.48/44.77/58.01	55.46
Gaussian RBF	**56.44/71.87/80.85**	63.84
Bump RBF	55.34/70.85/80.09	64.27

Table 6.6: Experiment 3A: Root identification and IPT reconstruction accuracies for iris images in the cross-modality setting.

Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
Legendre	56.75/76.58/87.88	67.53
Chebyshev	**72.59/88.29/94.90**	**67.90**
Gabor	5.79/12.40/19.70	51.23
Gaussian RBF	40.08/63.64/76.45	66.74
Bump RBF	39.67/60.88/76.03	66

Table 6.7: Experiment 3B: Root identification and IPT reconstruction accuracies for fingerprint images (Config -I) in the cross-modality setting.

Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
Legendre	29.66/44.32/56.82	68.99
Chebyshev	**31.93**/46.14/57.39	**70.59**
Gabor	22.50/37.27/51.36	68.08
Gaussian RBF	30.80/50.34/61.14	68.98
Bump RBF	31.59/**51.70/62.50**	68.51

Table 6.8: Experiment 3B: Root identification and IPT reconstruction accuracies for fingerprint images (Config -II) in the cross-modality setting.

Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
Legendre	34.58/51.11/59.31	65.82
Chebyshev	**35/55.28/65.28**	**65.93**
Gabor	31.39/48.75/63.33	60.76
Gaussian RBF	14.58/23.75/36.81	59.29
Bump RBF	17.50/28.33/40.42	59.96

Table 6.9: Experiment 3: Baseline performance of basis functions in terms of root identification and IPT reconstruction accuracies in the intra-modality setting.

Modality & Configuration	Basis Function	Root identification (%) Rank 1/2/3	IPT Reconstruction (%)
IRIS	Legendre	64.46/83.75/91.46	**68.62**
	Chebyshev	**76.31/89.53/94.63**	66.62
	Gabor	8.54/18.60/27.55	54.33
	Gaussian RBF	29.89/49.31/66.25	60.77
	Bump RBF	25.21/38.15/48.76	58.06
FINGERPRINT Config-I	Legendre	34.69/48.13/57.81	**71.92**
	Chebyshev	**38.75/58.44/67.55**	71.46
	Gabor	5.94/14.06/23.75	64.37
	Gaussian RBF	28.13/48.13/59.69	66.96
	Bump RBF	39.06/56.56/66.56	68.35
FINGERPRINT Config-II	Legendre	35.63/50.31/63.12	**66.22**
	Chebyshev	**42.50/56.56/69.69**	65.52
	Gabor	5.31/9.69/17.81	55.23
	Gaussian RBF	26.56/42.19/54.37	59.91
	Bump RBF	31.56/48.44/60.31	63.44

Table 6.10: Experiment 4: Root identification and IPT reconstruction accuracies for unseen photometric transformations.

Basis Functions	IPT Configurations							
	IPT 1		IPT 2		IPT 3		IPT 4	
	Root identification Rank 3 (%)	IPT Reconst-ruction (%)	Root identification Rank 3 (%)	IPT Reconst-ruction (%)	Root identification Rank 3 (%)	IPT Reconst-ruction (%)	Root identification Rank 3 (%)	IPT Reconst-ruction (%)
Legendre	**66.67**	82.22	**90**	71.67	**77.78**	44.44	**71.43**	**100**
Chebyshev	44.44	**84.44**	60	**75**	44.44	45.56	**71.43**	**100**
Gabor	44.44	82.22	80	71.67	**77.78**	**48.89**	**71.43**	**100**
Gaussian RBF	**66.67**	**84.44**	40	68.33	44.44	43.33	**71.43**	**100**
Bump RBF	**66.67**	**84.44**	60	68.33	33.33	44.44	**71.43**	**100**

(a) (b) (c)

Figure 6.10: Experiment 5: Example of geometric transformation (rotation) modeling using basis functions. (a) Original image (on the left) and the transformed image (on the right). (b) Modeled image pair using Legendre polynomials (modeled original image is on the left and modeled transformed image is on the right). (c) Modeled image pair using Gaussian RBF (modeled original image is on the left and modeled transformed image is on the right).

6.4.5 Results of Experiment 5

We present examples of geometric transformation modeling using the basis functions in Figure 6.10. We observe that Gaussian RBFs outperform Legendre polynomials at modeling the geometric transformations (see Figure 6.10) due to two reasons - (i) the radial basis functions can potentially span infinite range of values as opposed to the polynomials which can span values within a finite interval, and (ii) the RBFs did patch-level modeling compared to the pixel-level modeling done by the polynomials. The results in Table 6.11 indicate that the basis functions perform significantly better when trained on geometrically modified images (second protocol) compared to when trained on photometrically modified images (first protocol). As anticipated, if the class of transformations are the same in both training and testing set, the results are better, but surprisingly, even with photometrically modified training images, the basis functions are able to reliably handle geometric transformations. The basis functions outperform the baseline (see first row in Table 6.11) by $\sim 50\%$ in terms of root identification accuracy and $\sim 56\%$ in terms of IPT reconstruction accuracy. In the case of substituting the asymmetric measure in the baseline with the proposed asymmetric measure (we used Gaussian RBF), while retaining the tree spanning algorithm (see second row in Table 6.11), an improvement of $\sim 32\%$ in terms of root identification accuracy and an improvement of $\sim 52\%$ in terms of IPT reconstruction accuracy is observed.

We also observe that the IPT reconstruction accuracy is lower for geometrically modified images compared to photometrically modified images. We tried to further analyze this difference in performance. Visual inspection revealed that the geometrically modified images appeared ‘more

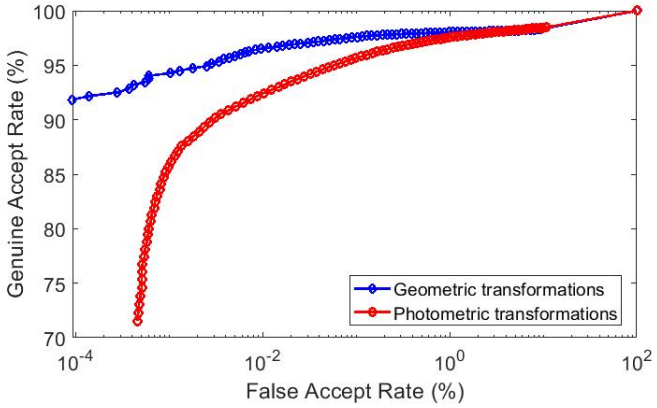


Figure 6.11: Experiment 5: ROC curves for recognition of the original images with the photometrically and geometrically modified images using a COTS face matcher. The recognition performance is higher for geometrically altered images compared to photometrically modified images indicating high degree of similarity with the original images.

similar' to the original image compared to the photometrically altered images. This can be attributed to the restrictive parameter range used in geometric transformations. A restrictive parameter range ensures that the images are indeed near-duplicates. A wide variation in the parameter values may result in highly dissimilar images, thereby, destroying the notion of near-duplicates. To quantify the degree of similarity between the original images and the modified images, we performed face recognition using a commercial face matcher. In the face recognition experiment, the original images served as the gallery and the geometrically modified images served as the probe samples. We repeat this same process for photometrically modified images belonging to Set I of Experiment 2. ROC curves presented in Figure 6.11 indicate a true match rate of 96.27% at a false match rate of 0.01% for the geometrically altered images, and a true match rate of 91.87% at a false match rate of 0.01% for the photometrically altered images. Note, the probe and gallery sizes for the photometrically modified images are four times more than that for the geometrically modified images. Nonetheless, the basis functions can better handle the identification of the original image (root node) and reconstruction of the IPT compared to the existing method.

Table 6.11: Experiment 5: Root identification and IPT reconstruction accuracies for geometric transformations. The top two rows indicate the baseline algorithms. The baselines yield only one root node as output so results are reported only at Rank 1 and the remaining ranks are indicated Not Applicable (NA). In this experiment, the testing (TE) is always done on geometrically modified images (indicated by TE-GM) but the training (TR) can be done using either photometrically modified images (indicated by TR-PM) or geometrically modified images (TR-GM). Results indicate training on geometrically modified images yield best performance when tested on geometric transformations.

Method	**Protocol**	**Root identification accuracy at Ranks 1/2/3 (%)**	**IPT reconstruction accuracy (%)**
Baseline	(SURF + MSAC) + Oriented Kruskal	8.00 / NA / NA	3.62
	Gaussian RBF + Oriented Kruskal	27.20 / NA / NA	7.24
Legendre	TR-PM, TE-GM	14.20 / 24.40 / 37.00	52.78
	TR-GM, TE-GM	23.20 / 39.60 / 52.80	54.30
Chebyshev	TR-PM, TE-GM	7.00 / 13.40 / 19.60	51.40
	TR-GM, TE-GM	23.20 / 35.60 / 49.20	**59.81**
Gabor	TR-PM, TE-GM	25.80 / 41.20 / 52.60	57.75
	TR-GM, TE-GM	25.60 / 40.00 / 54.60	58.49
Gaussian RBF	TR-PM, TE-GM	25.40 / 42.40 / 57.00	55.67
	TR-GM, TE-GM	**58.60 / 75.80 / 86.00**	51.40
Bump RBF	TR-PM, TE-GM	7.80 / 17.60 / 31.00	49.82
	TR-GM, TE-GM	46.60 / 65.80 / 77.00	56.32

6.4.6 Results of Experiment 6

Examples of the near-duplicates retrieved from the internet and their corresponding IPT reconstructions are presented in Figure 6.12. Qualitative analysis indicates that the reconstructed IPTs can depict the relationship between the near-duplicates reasonably well. For example, the images 3, 4 and 1 for the *Bob Marley* images (see Figure 6.12(c)) should ideally follow the sequence as indicated by the IPT. Similarly, image 2 appears to be a cropped version of image 3 which is correctly constructed by the proposed method.

6.4.7 Results of Experiment 7

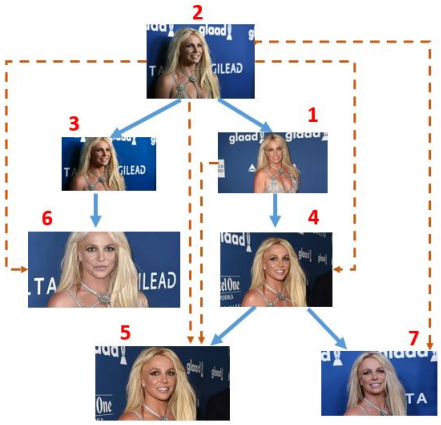
We used parameter distributions learnt from both photometrically modified images and geometrically modified images for IPT reconstruction. The results are reported in Tables 6.12 6.13 and in

(a) Bob Marley near-duplicates

(b) IPT constructed using Gabor

(c) IPT constructed using Gaussian RBF

(d) Britney Spears near-duplicates



(e) IPT constructed using Chebyshev

Figure 6.12: Experiment 6: Examples of near-duplicates available online and their corresponding IPTs constructed using the proposed method. The first row corresponds to (a) 4 near-duplicates retrieved using the query *Bob Marley*, (b) IPT constructed using Gabor trained on photometric distribution (the top 3 candidate root nodes are 2,3,1) and (c) IPT constructed using Gaussian RBF trained on geometric distribution (the top 3 candidate root nodes are 3,2,1). The second row corresponds to (d) 7 near-duplicates retrieved using the query *Britney Spears* and (e) IPT constructed using Chebyshev trained on photometric distribution (the top 3 candidate root nodes are 2,4,5). The bold arrows indicate immediate links and the dashed arrows indicate ancestral links.

terms of root identification and IPT reconstruction accuracies.

(a) h (b) (c)

Figure 6.13: Example images from the CelebA dataset containing prominent background details in the face images.

1. For the near-duplicates generated using autoencoder, the Gabor model performs the best in terms of root identification accuracy (Rank 1 accuracy of 81.25%) and the Chebyshev polynomials perform the best in terms of IPT reconstruction accuracy (64.37%). Gabor outperforms the remaining basis functions due to the characteristics of the CelebA dataset that includes face images along with with background details (see Figure 6.13). Gabor as a texture descriptor is able to accurately model the change in the texture of the background regions in the outputs reconstructed using the autoencoder.

2. For the near-duplicates generated using image augmentation package, maximum root identification accuracy of 50% at Rank 3 is obtained by Bump RBF while Chebyshev obtains an IPT reconstruction accuracy of 75.20%. The image augmentation packages include elastic distortions that are complex non-linear deformations, that the basis functions did not model accurately.

3. For the near-duplicates generated using the BeautyGlow network, we only performed qualitative evaluation due to the small size of the test set. We observe that the IPT constructed in Figure 6.6(b) indicates a deep tree with gradual increase in the intensity of the make-up which is anticipated.

We observed that the basis functions perform better when trained on photometrically modified images compared to when they are trained on geometrically modified images, in the case of both autoencoder and Augmentor. This is perhaps due to the fact that the autoencoder does not introduce geometric modifications; the modifications are restricted to structural and textural details. On the other hand, the image augmentation library produces random geometric modifications such as

Table 6.12: Experiment 7: Root identification and IPT reconstruction accuracies for deep learning-based transformations. In this case, the near duplicates are generated using autoencoder .

Basis functions	**For near-duplicates generated using autoencoder**			
	Performance when trained on photometric transformations		**Performance when trained on geometric transformations**	
	Root identification accuracy (%) @ Ranks 1/2/3	IPT reconstruction accuracy (%)	Root identification accuracy (%) @ Ranks 1/2/3	IPT reconstruction accuracy (%)
Legendre	25.00 / 50.00 / 68.75	50.63	25.00 / 50.00 / 62.50	62.50
Chebyshev	25.00 / 56.25 / 62.50	**64.37**	37.50 / 43.75 / 68.75	46.88
Gabor	**81.25 / 100 / 100**	55.00	31.25 / 43.75 / 56.25	49.38
Gaussian RBF	87.50 / 93.75 / 93.75	55.00	43.75 / 56.25 / 68.75	50.62
Bump RBF	56.25 / 68.75 / 87.50	57.50	25.00 / 37.50 / 43.75	55.00

Table 6.13: Experiment 7: Root identification and IPT reconstruction accuracies for deep learning-based transformations. In this case, the near duplicates are generated using image augmentation schemes for training deep neural networks.

Basis functions	**For near-duplicates generated using image augmentation schemes**			
	Performance when trained on photometric transformations		**Performance when trained on geometric transformations**	
	Root identification accuracy (%) @ Ranks 1/2/3	IPT reconstruction accuracy (%)	Root identification accuracy (%) @ Ranks 1/2/3	IPT reconstruction accuracy (%)
Legendre	16.00 / 27.00 / 40.00	74.00	16.00 / 30.00 / 42.00	71.27
Chebyshev	13.00 / 24.00 / 39.00	**75.20**	18.00 / 29.00 / 43.00	72.00
Gabor	12.00 / 25.00 / 36.00	71.93	18.00 / 30.00 / 40.00	70.27
Gaussian RBF	17.00 / 30.00 / 43.00	65.20	19.00 / 30.11 / 41.00	67.67
Bump RBF	**17.00 / 32.00 / 50.00**	66.20	12.00 / 26.00 / 44.00	67.40

rotation and re-sampling, but uses more sophisticated techniques to remove some of the artifacts associated with such operations (*e.g.*, removes the black padding near the borders of a rotated image). In such cases, we speculate that the basis functions view the geometrically altered image as a photometrically modified image. This explains the better performance of the basis functions when trained on photometrically modified images compared to geometrically modified images.

6.4.8 Further Analysis

In addition to the seven experiments discussed above, we performed another four experiments for further analysis: (i) to analyze the performance of the proposed algorithm on steganography, (ii) to evaluate the performance of the proposed method on handling arbitrary number of nodes (images), (iii) to analyze the impact of missing nodes on the performance of the method, and (iv)

to examine the impact of demographic influences on the method. To achieve the first task, we used near-duplicates generated using steganographic algorithm. Steganography refers to embedding a hidden message within the 'cover image' to generate a 'stego image'. Both cover and stego images have visually indiscernible differences, and the hidden message can be deciphered using dedicated steganalysis techniques. The amount of bits per pixel that can be distorted in the cover image to embed the secret message is referred to as the payload. Since, the cover images and stego images can be considered near-duplicates, we have demonstrated the performance of our image phylogeny tree (IPT) construction algorithm on a well known stegonagraphic algorithm called S-UNIWARD [87]. This refers to a universal distortion function for steganography in an arbitrary domain using directional filter banks. We used the S-UNIWARD tool[3] to create near-duplicate stego images using different payload values $\{0.1, \cdots, 0.9\}$. We then applied five different basis functions for identifying the root node (original image) and IPT reconstruction. For the remaining three tasks, we evaluated only using Chebyshev polynomial as it has consistently shown good performance across other experiments. To accomplish the second task, we constructed 100 IPTs, each IPT comprising 10, 20, 30, 40 and 50 nodes resulting in a total of 15,000 near-duplicates. To accomplish the third task, we randomly removed 20%, 40%, 60% and 80% of the nodes from 100 IPTs, each IPT comprising 20 nodes. Finally, to accomplish the fourth task, we selected 178 subjects from the UNCW MORPH (Academic) dataset[4] belonging to five different ethnic groups: 41 subjects belonging to Asian demographic group, 33 subjects belonging to Black demographic group, 38 subjects belonging to Hispanic demographic group, 29 subjects belonging to Asian demographic group and 37 subjects belonging to White demographic group, and created 178 IPTs comprising 10 nodes resulting in a total of 1,780 near-duplicates.

For the stego images demonstrated in Figure 6.14, the best performing basis function was the Bump radial basis function which identified the correct root node at Rank 2 and achieved 55.56% IPT reconstruction accuracy. Note that in this example, SSIM varied between 97.54%

[3]http://dde.binghamton.edu/download/stego_algorithms/

[4]https://ebill.uncw.edu/C20231_ustores/web/product_detail.jsp?PRODUCTID=8

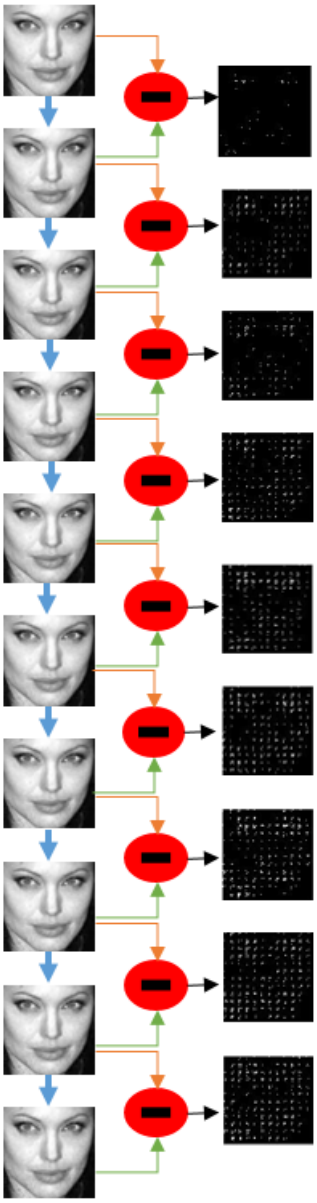


Figure 6.14: Illustration of steganographic images generated using the S-UNIWARD algorithm (on the left), and the differences in the coefficients in the DCT domain between the cover image and the stego image at each depth level (on the right).

and 99.81%. In spite of the extremely strong similarity between the ten images, our method was able to identify the correct root node with 66.67% accuracy, and recovered more than half of the edges, including ancestral and immediate links. For the task involving arbitrary number of nodes, we observed that the method can handle upto 20 nodes (75% root identification accuracy and 70% IPT reconstruction accuracy), but the structures become deeper after 30 nodes, and also the images start losing biometric utility due to severe modifications resulting in overall degradation in root identification and IPT reconstruction accuracy. For the task involving missing nodes, we observed the best performing results to be 85% in terms of root identification and 60% in terms of IPT reconstruction accuracy. Finally, as far as demographic influences are concerned, we observed that the root identification is slightly biased towards the 'White' group, possibly due to the training set involving LFW dataset which has a majority of White subjects. However, the IPT reconstruction accuracy was observed to be consistent across different demographic groups.

Next, we highlight the main take-away points from all the experiments.

1. Radial basis functions performed best at modeling the baseline photometric transformations (Brightness adjustment, Gamma transformation, Gaussian smoothing and Median filtering) and also geometric transformations. They resulted in the lowest residual error when used for modeling the transformations.

2. Orthogonal polynomials performed best at reliably discriminating between the forward and reverse directions. They resulted in the highest root identification and IPT reconstruction accuracies in a majority of the cases involving photometric and geometric modifications.

3. The proposed approach of utilizing "likelihood ratio" generalizes well across multiple IPT configurations and different biometric modalities.

4. The proposed approach is capable of handling different classes of transformations such as photometric and geometric. In addition, they are robust to unseen transformations using deep learning tools and image editing software.

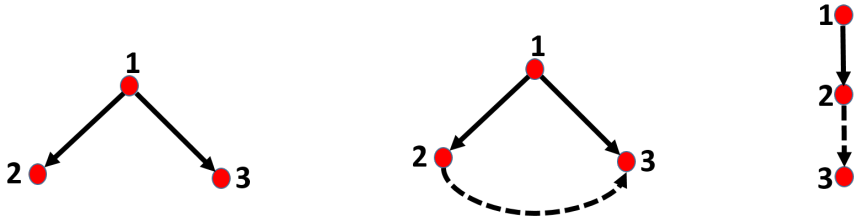


Figure 6.15: Toy example demonstrating the effect of insertion of spurious edge on the von Neumann entropy. (a) Groundtruth IPT (b) Correctly reconstructed IPT with spurious edge (c) Incorrectly reconstructed IPT with spurious edge. Note, the spurious edge is indicated by dashed line.

Table 6.14: Approximate von Neumann entropy for analysis of spurious edges and missing edges in reconstructed IPTs. The mean and the standard deviation of the differences between the entropy of the ground truth and the reconstructions are reported. Low values indicate accurate reconstructions and smaller number of spurious as well as missing edges.

Basis Function	**Entropy (Mean and standard deviation)**
Legendre6	-0.0009 ± 0.0045
Chebyshev	**-0.0000 ± 0.0044**
Gabor	-0.0036 ± 0.0039
Gaussian RBF	-0.0035 ± 0.0028
Bump	-0.0034 ± 0.0030

6.5 Explanatory Model

Finally, in this section, we present a graph entropy-based explanatory model which analyzes the failure cases of the IPTs reconstructed using the proposed method. A failed IPT reconstruction can involve (i) missing edges, and (ii) spurious edges. Such failure cases can be quantified using approximate von Neumann entropy computed for directed graphs. Graph entropy is computed in terms of its in-degree and out-degree [162]. Consider a directed graph $G(V, E)$ with a set of nodes denoted by V and the set of edges denoted by E. The adjacency matrix of such a graph is defined as,

$$A_{uv} = \begin{cases} 1 & \text{if } (u, v) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

The in-degree and out-degree of node u are presented as $d_u^{in} = \sum_{v \in V} A_{vu}; d_u^{out} = \sum_{v \in V} A_{uv}$. The von

Neumann entropy for the directed graph G can be written as:

$$H = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2}\left[\sum_{(u,v)\in E} \frac{d_u^{in}}{d_v^{in} d_u^{out^2}} + \sum_{(u,v)\in E_2} \frac{1}{d_u^{out} d_v^{out}}\right]. \tag{6.5.1}$$

Here, $E_1 = \{(u,v)|(u,v) \in E \text{ and } (v,u) \notin E\}$; $E_2 = \{(u,v)|(u,v) \in E \text{ and } (v,u) \in E\}$, such that, $E = E_1 \cup E_2$ and $E_1 \cap E_2 = \emptyset$. The maximum value of entropy for a directed graph is equal to $1 - \frac{1}{|V|}$ for a star graph where all the nodes in the graph have either incoming links or outgoing links but not both. The minimum entropy is obtained for a cyclic graph, implying all nodes are fully connected, and the value of minimum entropy is $1 - \frac{1}{|V|} - \frac{1}{2|V|^2}|V| = 1 - \frac{1}{|V|} - \frac{1}{2|V|} = 1 - \frac{1.5}{|V|}$. In our work, we want to move toward maximum entropy since minimum entropy results in the worst case scenario of reconstruction where the result is a cyclic graph. Thus, Eqn. (6.5.1) can be simplified as follows.

$$H = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2}\left[\sum_{(u,v)\in E_1} \frac{d_u^{in}}{d_v^{in} d_u^{out^2}}\right]. \tag{6.5.2}$$

The above equation is applicable in our work since $E_2 = \emptyset$, otherwise, we will end up having a cycle in the reconstructed IPT. We hypothesize that spurious edges decreases the entropy of the IPT, which is further reduced if the IPT misses correct edges. This can be illustrated using a toy example presented in Figure 6.15. The reconstruction yields a 100% IPT reconstruction accuracy for the first reconstructed IPT (Figure 6.15(b)), which has no missing edge but one spurious edge, and 50% for the second reconstructed IPT (Figure 6.15(c)), which has one missing edge and one spurious edge. The von Neumann entropy as computed from Eqn. (6.5.2) for the ground truth configuration (Figure 6.15(a)) is, $H(GT) = 1 - \frac{1}{3} - \frac{1}{18}\left[\frac{d_1^{in}}{d_2^{in} d_1^{out^2}} + \frac{d_1^{in}}{d_3^{in} d_1^{out^2}}\right] = 1 - \frac{1}{3} - \frac{1}{18}\left[0\right] = 0.67$. Note that the in-degree of node 1 is 0. This value also corresponds to the maximum entropy of a directed graph with 3 nodes. The entropy for the first reconstructed IPT (Figure 6.15(b)) is, $H(IPT1) = 1 - \frac{1}{3} - \frac{1}{18}\left[\frac{d_1^{in}}{d_2^{in} d_1^{out^2}} + \frac{d_1^{in}}{d_3^{in} d_1^{out^2}} + \frac{d_2^{in}}{d_3^{in} d_2^{out^2}}\right] = 1 - \frac{1}{3} - \frac{1}{18}\left[0 + 0 + \frac{1}{2 \times (1)^2}\right] = 0.64$. The entropy for the second reconstructed IPT (Figure 6.15(c)) is, $H(IPT2) = 1 - \frac{1}{3} - \frac{1}{18}\left[\frac{d_1^{in}}{d_2^{in} d_1^{out^2}} + \frac{d_2^{in}}{d_3^{in} d_2^{out^2}}\right] = 1 - \frac{1}{3} - \frac{1}{18}\left[0 + \frac{1}{1 \times (1)^2}\right] = 0.61$. Thus, $H(GT) > H(IPT1) > H(IPT2)$. This demonstrates that

inaccurate IPTs (missing edges and spurious edges) can reduce the entropy from the ground truth entropy, and this property can be leveraged for evaluating the goodness of the reconstructed IPTs. Reconstructed IPTs with missing edges and spurious edges will tend to have lower entropy.

We compute this entropy-based measure to analyze the accuracy of the IPTs reconstructed using the proposed method for the face images (full set with 2,727 IPTs). The maximum and minimum entropy for a graph containing 10 nodes is 0.85 and 0.90, respectively. The von Neumann entropy as computed from Eqn. (6.5.2) for the ground truth IPT (see Figure 6.2(a)) is 0.89. We then compute the entropy corresponding to the reconstructed IPTs. Finally, we compute the difference between the entropy of the ground truth configuration and the reconstructed configurations. We report the mean and the standard deviation of the differences in the entropy in Table 6.14. The results indicate that the differences between the ground truth entropy and the entropy of the reconstructed IPTs are the smallest when the polynomials are utilized as basis functions. This further corroborates our findings that the polynomials result in the best IPT reconstruction accuracies (see Tables 6.4 - 6.9).

6.6 Summary

In this chapter, we presented a method to model pairwise photometric and geometric transformations between a set of near-duplicate biometric images using five basis functions. The modeling of the transformations is used in conjunction with the likelihood ratio based asymmetric measure to identify the original image and deduce how the images are related to each other. We performed comprehensive experiments on a large dataset comprising 27,270 images belonging to face, iris and fingerprint modalities. We further conducted evaluations on unseen modalities and unseen transformations. The proposed method was robust against unseen modalities as well as unseen transformations accomplished using image editing software, steganography and deep learning-based methods. The proposed method capably handled arbitrary number of nodes (images), missing nodes and near-duplicates downloaded from the Internet reasonably well. We also analyzed the performance of the basis functions using visualization aid (t-SNE) and observed that some of the photometric transformations are easier to model than others. Finally, we utilized

von Neumann directed graph entropy to evaluate the reconstructed IPTs. The proposed algorithm outperformed existing state-of-the art methods by upto 37% in terms of root identification accuracy and by upto 47% in terms of IPT reconstruction accuracy.

CHAPTER 7

GRAPH-BASED APPROACH FOR IMAGE PHYLOGENY FOREST

7.1 Introduction

In this chapter, we propose a novel image phylogeny construction technique that employs sensor pattern noise and graph convolution technique to (re)construct an image phylogeny forest. The idea is to develop an unified framework that integrates sensor-specific details and image-specific details for the task of image phylogeny. An image phylogeny forest (IPF) comprises multiple phylogeny trees, and each tree consists of a distinct root node and can have arbitrary structure. This removes the earlier constraint of having a single root node, and that all images should be related. The task of IPF construction can be considered as an extension of image phylogeny tree (IPT) construction but with the reject class option. It implies that all the images may not typically belong to a single IPT. Instead, there can be cases when some images may be outliers (anomalies) and will remain as singletons, or may belong to a different IPT. To accomplish this objective of identifying which images belong to which IPT, we present a novel locally-scaled spectral clustering technique to identify the number of clusters (IPTs). Following the clustering process, we utilize sensor pattern noise (PRNU) with graph convolutional network to construct each IPT. By repeating this process, for all the clusters (IPTs) concerned, we achieve IPF construction. Also, this is different from our previous work, because the basis functions perform a pairwise analysis, while the current work performs both global and local analysis. It applies spectral graph theory to all the images simultaneously, performing a global analysis; followed by sensor pattern noise features to perform link prediction between pairs of nodes, thereby performing a local analysis.

In this chapter, the objective is as follows: **Given a set of near-duplicate face images, our objective is to construct an IPF.** See Figure 7.1. We propose a novel IPF construction method with two components: (i) An improved spectral clustering algorithm which uses locally-scaled kernels to address the multi-scale issue for clustering the images correctly. (ii) A graph-based approach

in conjunction with sensor noise pattern features extracted from the near-duplicate set in order to examine both local (pairwise) and global (all images simultaneously) interactions to accurately construct the IPT corresponding to each cluster. The **contributions** of this work are as follows:

1. We develop a locally-scaled spectral clustering technique using features extracted from near-duplicate images to identify and distinguish between images belonging to disparate IPTs. This step also helps in determining the number of IPTs in an IPF.

2. We develop a technique for IPT construction by combining a graph convolutional network (GCN) with sensor noise pattern features to harness the capacity of both local and global analyses. The GCN serves as a *node embedding* module to determine global relationships between near-duplicate images by identifying the hierarchical position of an image in the IPT. We use sensor noise pattern features to inspect local relationships for *link prediction* between the original image and the transformed image. By doing so, we have designed a method that leverages both degrees of analysis for an accurate determination of the relationships between the near-duplicates. Finding relationships between near-duplicates is not a trivial task as the transformations used to generate a child node from a parent node may not have a closed form representation or may have a vast range of parameter space making the modeling of the transformations extremely difficult, and most importantly there may not be a unique mapping from the parent image to the child image. Therefore, we need an approach that can perform a holistic analysis to reliably determine how the images are related to each other.

3. We evaluate the proposed clustering technique on simulated examples (near-duplicates generated using deep learning transformations) and real world examples (images downloaded from the Internet) and compare its performance with conventional spectral clustering.

4. We perform a rigorous analysis of the proposed IPT construction method (node embedding and link prediction) by evaluating on photometrically and geometrically modified images and across unseen transformations, unseen IPT configurations and across different biometric modalities.

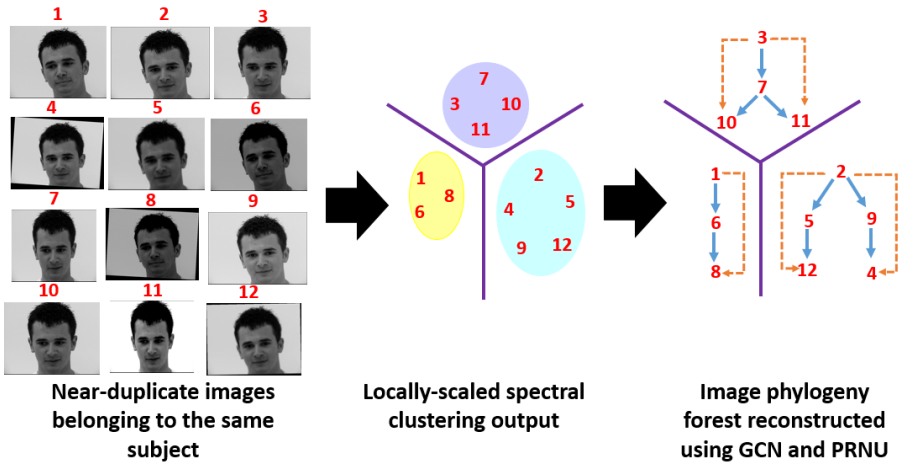


Figure 7.1: Outline of the objective in this work. Given a set of near-duplicate face images belonging to the same subject (near-duplicates can be generated using either photometric or geometric transformations or both), our objective is two-fold. Firstly, we would like to filter out the images that do not belong to the same evolutionary structure. We achieve this by using a locally-scaled spectral clustering step. The clusters indicated by ellipsoids vary in diameter indicating the importance of local scaling. Secondly, for each cluster, an image phylogeny tree (IPT) is constructed. The ensemble of IPTs result in the desired output corresponding to an Image Phylogeny Forest (IPF).

5. Finally, we evaluate their joint performances in the context of IPF construction for face images acquired using different cameras and with different expressions and compare with state-of-the art baseline methods.

7.2 Proposed Method

The proposed IPF construction method has been broadly outlined in Figure 7.1. The proposed method has two phases or steps - grouping and phylogeny. Grouping involves clustering the input near-duplicates into disjoint groups. Intuitively, this implies that images which should not belong to the same IPT must be assigned to separate groups or clusters. Several clustering algorithms [93] exist such as k-means [110], density based clustering [141], spectral clustering [126], etc. but they suffer from some limitations. For example, k-means requires input value of number of clusters k, and is sensitive to noise and outliers. On the other hand, density-based clustering methods although effective for arbitrary shaped clusters, is sensitive to the parameter selection. Spectral clustering conceptualizes data points as nodes in the graph and interprets the distances between data points as

walks on the graph lying on a non-linear manifold. Determining the clusters requires computing the distances between data points, and in turn, a non-linear dimensionality reduction by using the eigen vectors of the graph Laplacian. The steps of spectral clustering can be summarized as follows:

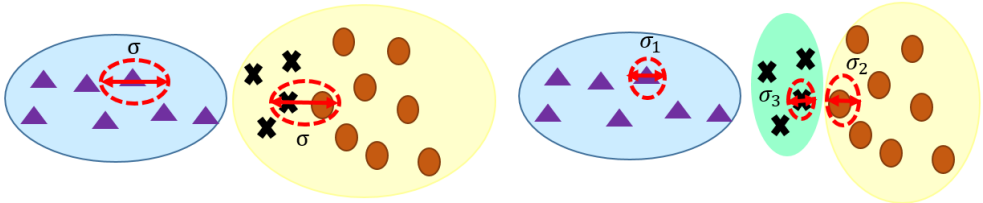
1. Construct a $Q \times Q$ symmetric similarity matrix $\boldsymbol{S}$ from the input data points $\boldsymbol{X} \in \mathcal{R}^{Q \times d}$ and its corresponding weighted adjacency matrix $\boldsymbol{A}$.

2. Compute the graph Laplacian matrix $\boldsymbol{L}$ from $\boldsymbol{A}$.

3. Perform eigen decomposition on $\boldsymbol{L}$ and select k eigen vectors corresponding to k smallest eigen values $\boldsymbol{U} = [u_1, u_2, \cdots, u_k]$, where $\boldsymbol{U} \in \mathcal{R}^{Q \times k}$ and each u_i is a k-dimensional column vector.

4. Apply k-means on the rows of $\boldsymbol{U}, \boldsymbol{U}(i, :)$ to obtain cluster assignment for each $\boldsymbol{X}_i, i = 1, \cdots, Q$.

Spectral clustering suffers from a major limitation that it is incapable of handling multi-scale cases as discussed in [29]. The notion of scale arises in the similarity matrix $\boldsymbol{S}$ construction used in the first step of spectral clustering. It uses a kernel, h (preferably a smooth kernel such as an exponential decay function) to compute the affinity between a pair of points $(\boldsymbol{X}_i, \boldsymbol{X}_j)$ as $\boldsymbol{S}(i, j) = h\left(\frac{\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2}{\sigma^2}\right)$. The parameter σ refers to the bandwidth variable, and is usually computed by training on a large number of data points. A correct computation of σ is pivotal for accurate clustering but is governed by the implicit assumption that all the clusters have similar scales, *i.e.*, approximate uniform distribution of data points across all the clusters. This will result in inaccurate clustering when the number of data points within a cluster varies. As shown in Figure 7.2(a), a single global bandwidth results in two clusters because the bandwidth considers the data points 'x' and 'o' to be grouped together. An effective strategy to deal with the multi-scale issue is to consider locally-scaled kernels which vary with the data points and is therefore invariant to sampling density [29]. Then the similarity matrix can be formulated as $\boldsymbol{S}(i, j) = h\left(\frac{\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2}{\sigma(\boldsymbol{X}_i)\sigma(\boldsymbol{X}_j)}\right)$. The variables $\sigma(\boldsymbol{X}_i)$ and $\sigma(\boldsymbol{X}_j)$ are known as local bandwidths as they are locally tuned to the data point under consideration. As a result the data points in multi-scale clusters are grouped correctly



(a) (b)

Figure 7.2: Illustration of the proposed spectral clustering which uses locally-scaled kernels (bottom) instead of a single kernel with a global bandwidth σ (top). The number of images in each cluster, *i.e.*, the density of each cluster is not known *apriori*. The global bandwidth incorrectly merges two clusters. On the other hand, the local scales (σ_1, σ_2 and σ_3) are computed assuming that clustering is inherently a geometric problem resulting in three correct clusters in this example.

as depicted in Figure 7.2(b). This can be rationalized by leveraging a geometric interpretation of spectral clustering. Spectral clustering can be treated as a technique "for representing function spaces on a manifold" [29].

7.2.1 Locally-scaled spectral clustering

The multi-scale limitation posed by conventional spectral clustering concerns our work as we do not assume the scale or the size of the cluster apriori, *i.e.*, the number of nodes (images) in each IPT. It may be that some IPT have 3 nodes, whereas another IPT may have 15 nodes. Such wide variations will impair the performance of the spectral clustering (also shown empirically later), and will lead to an overall poor performance in IPF construction. Taking the geometric interpretation of spectral clustering into account, we propose *locally-scaled* spectral clustering, where we apply the local-scaling on the feature space ($f(\boldsymbol{X})$) instead of directly applying it on the data space ($\boldsymbol{X}$). One of the advantages of applying it on the feature space is that it will allow aggregation of multiple features that can produce an accurate similarity matrix, and subsequently accurate clustering results. We selected three features in this work: (i) pixel intensity, (ii) sensor noise pattern features, particularly Enhanced Photo Response Non-Uniformity (PRNU) [106], and (iii) face descriptors. The choice for selecting pixel intensity and PRNU features stem from the assumptions that the near-duplicates may have been taken in different settings (indoor or outdoor) or using different cameras. Therefore,

pixel intensity can help discern between images depending on illumination variations, whereas, PRNU which contains sensor-specific details can help disambiguate between images captured using different cameras. Lastly, as we are using face images which may vary in expression or pose, a face descriptor may help distinguish between near-duplicates belonging to different evolutionary sequences. We obtained face descriptors using a neural network architecture such as VGGFace [132] in this work. We can always substitute face descriptors with generic image descriptors for images depicting natural scenes. The intuition is to harness the capability of complimentary features to augment the construction of an accurate weighted symmetric similarity matrix $\boldsymbol{S}$. We use a smooth exponential decay function as our kernel ($h(\cdot)$) for computing the similarity matrix as suggested in [29]. The resultant similarity matrix now becomes

$$\boldsymbol{S}(i,j) = \exp -\left(\frac{\boldsymbol{D}_F(i,j)^2}{\hat{p}_F(i)\hat{p}_F(j)} + \frac{\boldsymbol{D}_N(i,j)^2}{\hat{p}_N(i)\hat{p}_N(j)} + \frac{\boldsymbol{D}_P(i,j)^2}{\hat{p}_P(i)\hat{p}_P(j)} \right)$$

Here, $\boldsymbol{D}_F(i,j) = 1 - \frac{\boldsymbol{F}_i \boldsymbol{F}_j}{\boldsymbol{F}_i^2 \boldsymbol{F}_j^2}$, where, $F(\cdot)$ is the face descriptor extracted from the image; $\boldsymbol{D}_N(i,j) = 1 - \frac{\boldsymbol{N}_i \boldsymbol{N}_j}{\boldsymbol{N}_i^2 \boldsymbol{N}_j^2}$, where $N(\cdot)$ is the PRNU extracted from the image; $\boldsymbol{D}_P(i,j) = 1 - \frac{\boldsymbol{P}_i \boldsymbol{P}_j}{\boldsymbol{P}_i^2 \boldsymbol{P}_j^2}$, where, $P(\cdot)$ is the pixel intensity features extracted from the image. The terms in the denominator are locally-scaled bandwidths computed using univariate kernel density estimate of the respective features. $\hat{p}_F(\cdot)$ corresponds to kernel density estimate (KDE) computed from face descriptors, $\hat{p}_F(\cdot) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{F(\cdot) - F(x_i)}{h} \right)$, where n is the number of data points, h is the bandwidth equal to the interval between all the points in x and K is a normal kernel. Similarly, $\hat{p}_N(\cdot)$ denotes KDE computed from the PRNU features, and $\hat{p}_P(\cdot)$ denotes KDE computed from the pixel intensity features. Next, we compute binarized similarity matrix $\boldsymbol{S}_B$ using median of each row as threshold, such that $\boldsymbol{S}_B(i,j) = 1$ if $\boldsymbol{S}(i,j) > median(\boldsymbol{S}(i,:))$, and $\boldsymbol{S}_B(i,j) = 0$, otherwise. Then we follow the steps pertaining to conventional spectral clustering, such as computation of the degree matrix, Laplacian matrix, and eigen decomposition. The degree matrix is computed as $\boldsymbol{Deg}(i,i) = \sum_{j=1}^{Q} \boldsymbol{S}_B(i,j)$. The normalized Laplacian is computed as $\boldsymbol{L} = \boldsymbol{Deg}^{-\frac{1}{2}} \boldsymbol{S}_B \boldsymbol{Deg}^{-\frac{1}{2}}$. Eigen value decomposition of the Laplacian matrix results in eigen vectors ($\boldsymbol{U}$) and eigen values (Λ). Next, we apply a threshold η (selected using a validation set) to obtain a subset of eigen values, say k smallest eigen values such

that, $\{\Lambda_k \subset \Lambda | \Lambda_k < \eta\}$. The corresponding eigen vectors become $\boldsymbol{U}_k = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_k]$. Finally, k-means is applied to the rows of $\boldsymbol{U}_k$ to get cluster assignments, $C \in \mathcal{R}^{Q \times 1}$ for each image, such that $C = \text{k-means}(\boldsymbol{U}_k, k)$, where, k is the number of clusters. The details of the method are outlined in Algorithm 7.

Algorithm 7: Locally-scaled spectral clustering

Input: Set of Q near-duplicate images $\boldsymbol{I}$
Output: Cluster assignments C
Compute the face descriptors $\boldsymbol{F}$ from images (we used VGGFace)
Compute the PRNU $\boldsymbol{N}$ from images (we used Enhanced PRNU)
Compute the vectorized representation of the pixel intensities $\boldsymbol{P}$ from images
Compute kernel density estimate for face descriptors, PRNU features and pixel features, respectively: $\hat{p}_F(\cdot), \hat{p}_N(\cdot), \hat{p}_P(\cdot)$
while $i, j \leq Q$ **do**
    Compute distance values between each pair of images $(\boldsymbol{I}_i, \boldsymbol{I}_j)$ for face descriptors, PRNU features and pixel features, respectively (we used cosine distance) $\boldsymbol{D}_F(i,j) = 1 - \frac{\boldsymbol{F}_i \boldsymbol{F}_j}{\boldsymbol{F}_i^2 \boldsymbol{F}_j^2}$; $\boldsymbol{D}_N(i,j) = 1 - \frac{\boldsymbol{N}_i \boldsymbol{N}_j}{\boldsymbol{N}_i^2 \boldsymbol{N}_j^2}$; $\boldsymbol{D}_P(i,j) = 1 - \frac{\boldsymbol{P}_i \boldsymbol{P}_j}{\boldsymbol{P}_i^2 \boldsymbol{P}_j^2}$.
    Compute symmetric similarity matrix $\boldsymbol{S}$: $\boldsymbol{S}(i,j) = \exp - \left(\frac{\boldsymbol{D}_F(i,j)^2}{\hat{p}_F(i)\hat{p}_F(j)} + \frac{\boldsymbol{D}_N(i,j)^2}{\hat{p}_N(i)\hat{p}_N(j)} + \frac{\boldsymbol{D}_P(i,j)^2}{\hat{p}_P(i)\hat{p}_P(j)} \right)$
end while
Compute binarized similarity matrix $\boldsymbol{S}_B$ using median of each row as threshold
Compute diagonal degree matrix $\boldsymbol{Deg}$
Compute normalized Laplacian $\boldsymbol{L}$
Perform eigen value decomposition to obtain eigen vectors ($\boldsymbol{U}$) and eigen values (Λ) from Laplacian
Select thereshold η to select k smallest eigen vectors Λ_k: $\boldsymbol{U}_k = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_k]$
Apply k-means clustering on the rows of $\boldsymbol{U}_k$ to get cluster assignments C for each image **return** C

Upon obtaining the clusters using the proposed locally-scaled spectral clustering, we then proceed to the second phase of our IPF pipeline, *i.e.*, individual IPT construction. We have discussed different IPT construction strategies earlier but they all focus on either pairwise analysis of the images or global analysis of the entire set. In contrast, we propose a novel method that couples graph-based convolutional network (GCN) to perform a macroscopic analysis of the entire set and sensor noise pattern features to explore microscopic analysis of the images. In order to develop a global analysis of the near-duplicates, we must first define what global or macroscopic relationships

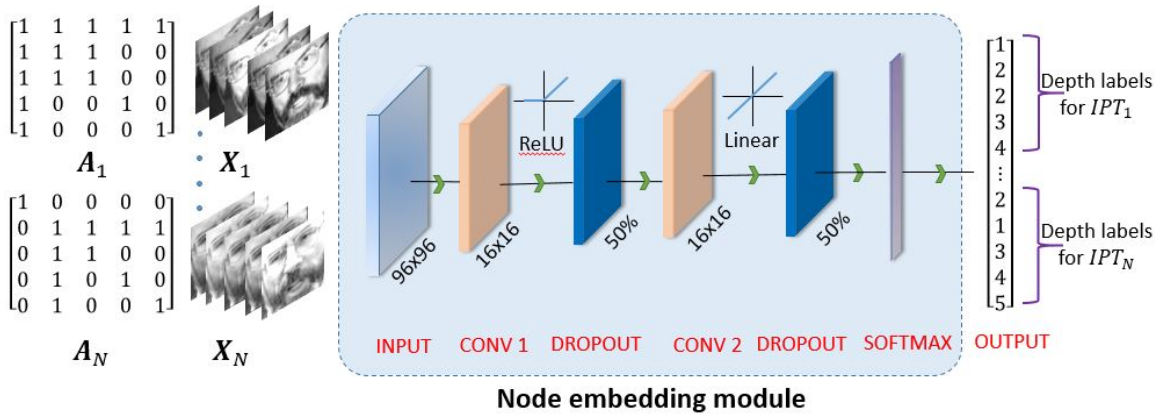


Figure 7.3: Illustration of the 'node embedding' module (Section 7.2.2). The module $f(\boldsymbol{X}, \boldsymbol{A})$ accepts a pair of inputs, $\boldsymbol{X}$: pixel intensity values of each image in the IPT and $\boldsymbol{A}$: an adjacency matrix indicating relationships between the images in the IPT. The output of this module is a vector of depth labels corresponding to each IPT configuration fed as input.

mean in the context of an IPT. An IPT represents a hierarchical structure, where each image is located at a depth k. The depth of an image in an IPT signifies the number of ancestors $(k-1)$ for that image, with respect to the root node, which is located at depth=1. For example, an image at depth=2 has only 1 ancestor, *i.e.*, the root node, while an image at depth=3 has 2 ancestors, and so on. The determination of the depths of the images in an IPT will provide a holistic understanding of how the images are *globally related* with respect to the remaining images. Therefore, the first step in our proposed approach is determining the *depth labels* for a set of near-duplicate images that are related to each other. We use a 'node embedding' module to accomplish the task of deducing the depth labels. Once the depth labels are computed for each image, we will determine how nodes (images) at successive depths are related to each other. For example, the root node will have only outgoing links, sibling nodes will not have any links between them, and leaf nodes will have only incoming links. We use a 'link prediction' module that utilizes sensor pattern noise (PRNU) to accomplish the task of identifying existence of links between nodes located at depth k and depths $> k$ by performing the microscopic analysis. The details of both modules are described next.

7.2.2 GCN-based Node Embedding

The task of this module is to accept a set of related near-duplicate images $\boldsymbol{I} \in \mathcal{R}^{M \times d \times d}$ and output a vector of depth labels $\boldsymbol{l} \in \mathcal{R}^{M}$. Here, M refers to the number of near-duplicate images belonging to a set (number of images belonging to an IPT or cluster which is less than Q, the total number of images we began with prior to clustering), and each image is of size $d \times d$. Let $f(\cdot, \cdot)$ represent the node embedding module which requires a pair of inputs $\boldsymbol{X}$ and $\boldsymbol{A}$. Therefore, $\boldsymbol{l} = f(\boldsymbol{X}, \boldsymbol{A})$. Here, $\boldsymbol{X}$ refers to the pixel intensity values of images $\boldsymbol{I}$, $\boldsymbol{X} \in \mathcal{R}^{M \times d^2}$. $\boldsymbol{A}$ represents the adjacency matrix of size $M \times M$ that represents some relationship between the images. The adjacency matrix can be computed using a nearest-neighbor based method utilizing features extracted from the images. We will discuss about the construction of the adjacency matrix in Section 7.3. Note that the adjacency matrix does not necessarily encode the IPT structure. See Figure 7.3.

Node embedding associates a node (an image) with a corresponding label (depth in the IPT). Graph-based convolutional networks employ spectral convolutions in the Fourier domain to model pairwise as well as higher order correlations in the data. They have been successfully used for semi-supervised node classification [102] and representation learning [81]. In this work, we studied three graph based neural networks, viz., (i) Graph Convolutional Network (GCN) [55], (ii) Graph Convolutional Network with Linear Approximation (GCN-Linear) [102] and (iii) Hypergraph Neural Network (HGNN) [67] to accomplish the task of node embedding. The underlying principle of a graph convolutional network is to perform spectral convolutions, in contrast to spatial convolutions, used in traditional convolutional neural networks. The spectral convolution filter is approximated using a truncated Chebyshev polynomial expansion of order K. The approximation reduces the computation complexity, as well as provides spatial localization, absent in the case of spectral convolution filter. The reader is referred to [55] for a detailed derivation. Setting the order of the polynomial $K = 1$, results in the variant known as GCN-Linear [102]. The linearization allows stacking of multiple convolutional layers to build deeper models. Finally, we use a hypergraph neural network [67] that allows hyperedges. The hyperedges combine multiple features extracted from the images *simultaneously* to determine high level correlations. This is in contrast to traditional

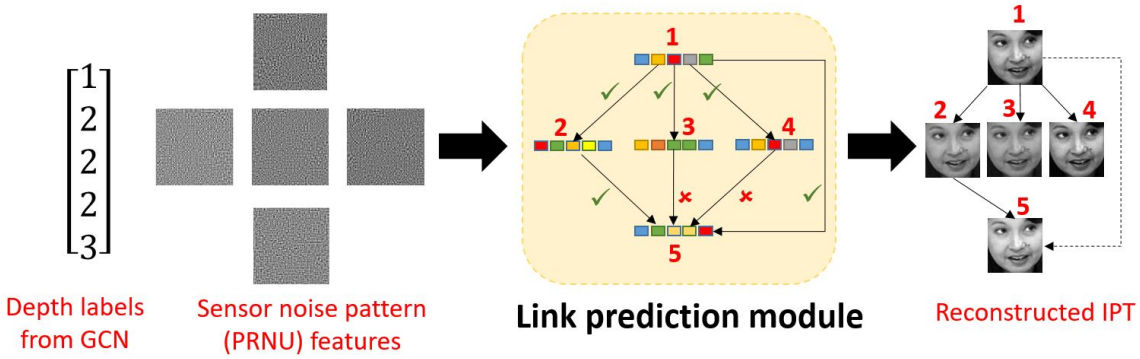


Figure 7.4: Illustration of the 'link prediction' module (Section 7.2.3). The module accepts a pair of inputs $g(\boldsymbol{l}, \boldsymbol{N})$, $\boldsymbol{l}$: depth labels from the 'node labeling' module (see Figure 7.3), and $\boldsymbol{N}$: sensor noise pattern (PRNU) features computed from each image of the set fed as input. The output of this module is the image phylogeny tree (IPT) containing edges directed from parent nodes to child nodes. Note that the ancestral links are present in the reconstructed IPT.

graph based networks that typically use a single feature at a time. HGNN is developed on top of GCN-Linear ($K = 1$) and uses multiple convolutional layers to represent multiple features for the task of node embedding.

7.2.3 PRNU-based Link Prediction

The task of this module is to accept the vector of depth labels $\boldsymbol{l} \in \mathcal{R}^M$ for a set of M near-duplicate images and output a data structure containing a set of vertices V, such that $|V| = M$, and directed edges E. We refer this directed structure as the image phylogeny tree: $\boldsymbol{IPT}(V, E)$. Let $g(\cdot, \cdot)$ represent the link prediction module that requires a pair of inputs $\boldsymbol{l}$ and $\boldsymbol{N}$. Therefore, $\boldsymbol{IPT} = g(\boldsymbol{l}, \boldsymbol{N})$. Here, $\boldsymbol{l}$ refers to the depth labels computed using the node embedding step. $\boldsymbol{N}$ represents the sensor noise pattern features computed from the images $\boldsymbol{I}$, such that, $\boldsymbol{N} \in \mathcal{R}^{M \times d^2}$. The depth labels computed by the node embedding step do not indicate how nodes at a particular depths are linked with nodes ate lower depths. Multiple nodes can have an identical depth label (see Figure 7.4), say depth label = k, but only one of them will be the parent of a node located at depth label $k + 1$. Additionally, there might some missing depth labels, in cases of incorrect outputs produced by the node embedding module (for example, multiple nodes with depth label=1). Therefore, the link prediction performs two tasks: (i) depth label correction, and (ii) pairwise link

inference.

Algorithm 8: PRNU-based link prediction

1: **Input:** Set of M near-duplicate images $\boldsymbol{I}$, set of depth labels $\boldsymbol{l}$ provided by GCN, where $|\boldsymbol{l}| = M$
2: **Output:** Root R and $\boldsymbol{IPT}$
3: Extract the sensor noise pattern (PRNU) for each image $\boldsymbol{N}(i)$ where $i = 1, \cdots, M$
4: Identify whether multiple nodes have depth label = 1, if $|Candidateroots| > 1$ go to Step 3, else go to Step 4
5: Perform depth label correction
6: Infer pairwise links
7: Identify the root node as the node with corrected depth label = 1, R
8: Construct the phylogeny tree: $\boldsymbol{IPT}(V, E)$ ▷ V represents the set of nodes and E represents the set of directed edges
return R and $\boldsymbol{IPT}$

In order to achieve the first task, *i.e.*, correcting missing depth labels, we first check whether multiple nodes have depth label=1, this is important because according to the definition of an IPT, it has only one root node (*i.e.*, only one node will have depth label=1). Multiple root nodes exist if $|Candidateroots| > 1$, where, $Candidateroots = arg_i\{\boldsymbol{l}(i) == 1\}$, where $i = 1, \cdots, M$. We employed sensor noise pattern features present in images for depth label correction. Photo Response Non-Uniformity (PRNU) is a type of sensor pattern noise which manifests in the image as a result of the non-uniform response of the pixels to the same light intensity [112], and can be used for sensor identification. Photometric and geometric transformations can induce changes in the PRNU pattern present in an image [19, 112]. See Figure 7.5 to visualize how PRNU patterns change in presence of transformations. We used the power spectral density (PSD) plots to demonstrate that, although the images may not depict significant variation, but their PRNU patterns exhibit changes and can therefore be utilized for link prediction, as it can help successfully discriminate between parent and child nodes. So, we hypothesize that PRNU can be used to correctly deduce the node which should have depth label=1. To accomplish that, we first compute the Euclidean distance between the PRNU features of each of the candidate root nodes and the remaining nodes, $\boldsymbol{D}_c = \|(\boldsymbol{N}(Candidateroots(c)) - \boldsymbol{N}(z))\|_2^2$, where $1 \leq z \leq M, z \neq c$, $\boldsymbol{N}$ corresponds to PRNU for each image. Next, we retain that node which results in the highest distance as the *correct* root node, $l(c) = 1$, where, $c = argmax(\boldsymbol{D})$. The rationale behind retaining the node with highest distance

as the node with depth label=1, is because as the depth increases, it implies that the root node (original image) has undergone multiple sequences of transformations, which will subsequently lead to higher variation in PRNU patterns of the transformed images. As a consequence, the root node will intuitively have the highest distance in terms of PRNU features between itself and the remaining nodes in the set. The nodes that were misclassified as depth label=1 were then re-assigned to depth label=2. After the depth label correction procedure, we proceed to the second task to infer the links and construct IPT.

In order to achieve the second task, *i.e.*, determine the existence of links between nodes located at depth labels k and $> k$, we again use the PRNU features. We will use the same notation $\boldsymbol{N}(i)$ to indicate PRNU of each image $i = \{1, \cdots, M\}$. Next we consider nodes located between successive depths, say, for example, nodes r and s that are assigned the same depth label, k, by the node embedding module, and another node t that is assigned depth label $k + 1$. We have to identify the correct parent (r or s) of the node t. For that, we compute the squared L_2- norm between their PRNU features: $D_{rt} = \|(\boldsymbol{N}(r) - \boldsymbol{N}(t))\|_2^2$ and $D_{st} = \|(\boldsymbol{N}(s) - \boldsymbol{N}(t))\|_2^2$. Finally, we select that node as the parent of t which results in the least distance, *i.e.*, $r \rightarrow t$ (r is the parent) if $D_{rt} < D_{st}$; otherwise, $s \rightarrow t$ (s is the parent). The rationale behind this decision is that unrelated nodes or ancestors are more likely to result in higher Euclidean distance with respect to their PRNU feature vector, whereas, the immediate parent will result in lower distance. We repeat this step for nodes located at successive depths until we have reached the leaf node(s). Finally, we deduce the root node as the node with corrected depth label = 1, $R = \arg(\boldsymbol{l}_{corr}(i) == 1), 1 \leq i \leq M$, where, $\boldsymbol{l}_{corr}$ represents the set of corrected depth labels. The steps of PRNU-based link prediction are summarized in Algorithm 8.

7.3 Implementation

The implementation of the GCN and GCN-Linear is based on [102].[1] The parameters used are as follows: learning rate = 0.01, number of epochs = 100, number of units in hidden layer = 16,

[1] https://github.com/tkipf/gcn

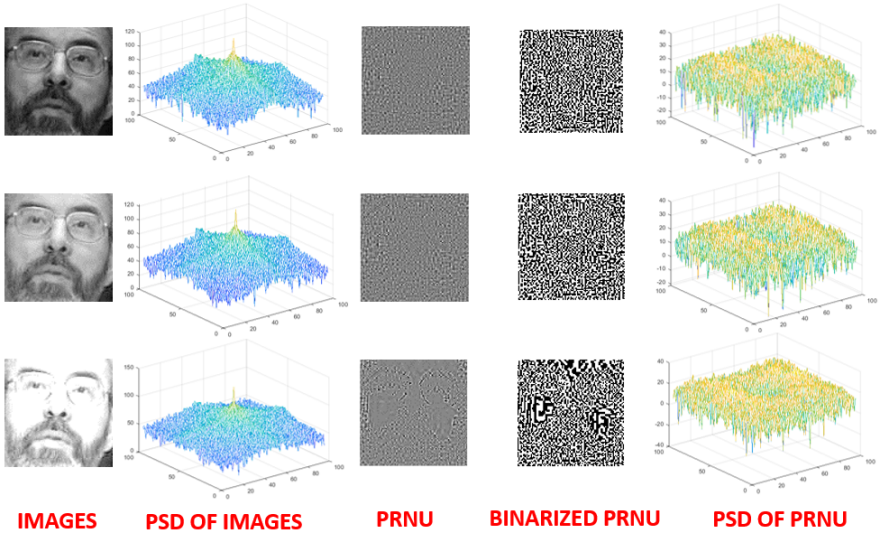


Figure 7.5: Illustration of utility of PRNU in image phylogeny. The graphic illustrates the variation in PRNU patterns in repsonse to photometric transformations. These variations are better visualized using the binary maps computed from each PRNU pattern (threshold=0) and their corresponding power spectral density (PSD) plots. Note, the PSD plots of the images do not bear any apparent variation, but the PSD plots of PRNU patterns reveal discernible differences. We intend to leverage this property of PRNU for the task of image phylogeny in conjunction with GCN.

dropout = 0.5, weight decay = 5×10^{-4}, early stopping = 10 (iterations) and degree of Chebyshev polynomial, $K = \{3, \cdots, 9\}$ (for GCN only). The cross-entropy loss function is employed. Both GCN and GCN-Linear require a feature matrix and an adjacency matrix. In our case, either the pixel intensity values or the PRNU features extracted from the images are used as the feature matrix of size $M \times D^2$, where, M is the number of images (in each set) and each image from the set is of size $D \times D$. For extracting the PRNU features, we used the method described in [106]. The method accepts a block diagonal matrix as the adjacency matrix input; each block represents a single $M \times M$ adjacency matrix corresponding to one set of near-duplicates. During *training*, we used the actual IPT configuration to construct the adjacency matrix. An asymmetric matrix can be used as adjacency matrix provided correct normalization is used for computing the graph Laplacian ($\boldsymbol{D}^{-1}\boldsymbol{A}$ instead of $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{A}$ is the adjacency matrix). Note that this degree matrix and adjacency matrix (for each near-duplicate set) are *different* from the ones used in locally-scaled spectral clustering (multiple near-duplicate sets). During *testing*, we do not assume any prior IPT configuration or image transformation. For a set of test images,

Table 7.1: Photometric and geometric transformations and the range of the corresponding parameters used in Experiments 2 and 3. The transformed images are scaled to $[0, 255]$. Note that these transformations are being used only in the training stage. For the test stage, any arbitrary transformation can be used.

Transformations	**Level of Operation**	**Parameters**	**Range**
Brightness adjustment	Global	[a,b]	a ∈ [0.9,1.5], b ∈ [-30,30]
Median filtering	Local	size of window [m,n]	m ∈ [2,6], n ∈ [2,6]
Gaussian smoothing	Global	standard deviation	$\sigma \in [1,3]$
Gamma transformation	Global	gamma	$\gamma \in [0.5,1.5]$
Translation	Global	$[T_x, T_y]$	$T_x \in [5,20]$, $T_y \in [5,20]$
Scaling	Global	Percentage	[90%, 110%]
Rotation	Global	theta	$\theta \in [-5^0, 5^0]$

first we compute their respective PRNU features, next we compute the squared L_2-norm between all PRNU feature pairs, and assign a link between the node pair if the distance is less than some threshold (mean is selected as threshold).

For HGNN,[2] the parameters used are as follows: learning rate = 0.001, number of epochs = 600, number of units in hidden layer = 128, dropout = 0.5, weight decay = 5×10^{-4}, multi-step learning rate scheduler parameters: gamma = 0.9, decay step = 200, decay rate = 0.7, milestones = 100 and the cross-entropy loss function is employed. We have used pixel features and PRNU features separately but we observed best results when both features are used together. The hypergraph adjacency matrix, $\in \mathcal{R}^{M \times 2M}$, is constructed by concatenating horizontally the adjacency matrices corresponding to the pixel features and PRNU features, when each of them is used in turn. Each adjacency matrix is constructed using the $k-$nearest neighbor ($k = 5$) method applied to the pixel and PRNU features, respectively.

7.4 Datasets and Experiments

We performed three sets of experiments:

1. Evaluation of the proposed locally-scaled spectral clustering in the context of deep learning-based transformations and images downloaded from the Internet.

[2]https://github.com/iMoonLab/HGNN

2. Evaluation of the proposed node embedding and link prediction module in the context of IPT reconstruction for photometrically and geometrically transformed images; in the context of unseen transformations; and in the context of unseen modalities and configurations.

3. Evaluation of the proposed clustering and IPT reconstruction method as a unified module for IPF reconstruction.

7.4.1 Experiment 1: Evaluation of locally-scaled spectral clustering

In this experiment, we used Google search query such as *Angelina Jolie* and *Superman* to download near duplicates from the **Internet**. These images are acquired in the wild with wide variations in poses and no sensor information. We also used 22 images belonging to four subjects from [84], where **deep learning**-based manipulations are applied on images to alter attributes such as adding hair bangs or adding glasses. Fine modifications can be applied such as adding left bangs or right bangs or making the shades more darker. These subtle modification result in near-duplicates. We have applied locally-scaled spectral clustering on these images to discern how many IPTs are present in an IPF.

7.4.2 Experiment 2: Evaluation of the proposed IPT reconstruction algorithm using GCN-based node embedding and PRNU-based link prediction

In this experiment, we evaluate the performance of the proposed approach in terms of (i) root identification accuracy, which computes the proportion of correctly identified root nodes, and (ii) IPT reconstruction accuracy, which is computed as follows: $\frac{|Original\_edges \cap Reconstructed\_edges|}{|Original\_edges|}$ for face images that are subjected to **photometric** and **geometric** transformations. We used face images from the Labeled Faces in the Wild (LFW) dataset [88]. All the images are cropped to a fixed size of 96×96 using a commercial face SDK. We used seven transformations—four photometric: Brightness adjustment, Gamma transformation, Median filtering and Gaussian smoothing, and three geometric: Rotation, Scaling and Translation. The parameters used in generating the near-duplicates are described in Table 7.1. 7,500 images from 123 subjects resulting in 1,500 IPTs

form the training set. 3,000 images from 46 subjects resulting in 600 IPTs form the validation set. We compared the performance of three node embedding techniques used in this work viz., GCN,

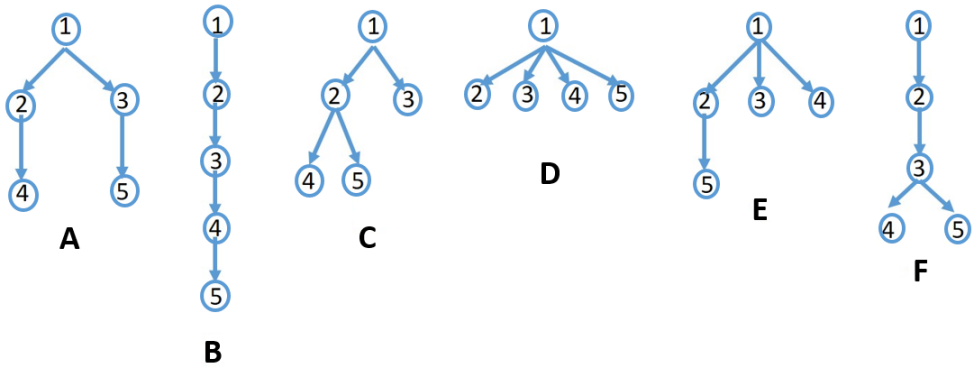


Figure 7.6: IPT configurations (structures) used in Experiment 2. For ease of visualization, only the immediate links are depicted. However, the ancestral links are also included for evaluation.

GCN-Linear and HGNN, and selected the network yielding the highest depth classification accuracy. The node labels (depth values) from the best performing model in the first stage are further fed to the link prediction module to infer the links for the six different IPT configurations depicted in Figure 7.6. There can be more than $n^{(n-2)}$ IPTs with n nodes (Cayley's formula). We selected these six configurations for training as they cover maximum breadth and depth values possible for an IPT with five nodes. We conducted experiments in two scenarios. In Scenario 1, we trained and tested on face images from the LFW dataset. In this scenario, the test set comprises 900 IPTs involving 4,500 face images corresponding to 75 subjects disjoint from the training and validation sets, and are evaluated separately for photometric and geometric transformations. In Scenario 2, we trained GCN on face images but tested on images from the Uncompressed Color Image Database (UCID) [142] depicting natural scene and generic objects. We used 50 images as used in [61] and applied photometric and geometric transformations to simulate 50 (number of original images)×5 (number of images in each IPT)×6 (number of IPT configurations)×2 (photometric and geometric transformations) = 3,000 test images. See Figure 7.7(b). For IPT reconstruction, only the GCN-based node embedding module requires training. Therefore, we wanted to analyze the robustness of the GCN module in handling different training and testing datasets.

Next, we performed an experiment to evaluate the performance of the proposed method in

the context of **unseen transformations**. We used Photoshop to manually edit face images from the LFW dataset resulting in a set of 175 near-duplicates. We generated the near-duplicates corresponding to 35 IPTs having 4 different configurations, and each configuration has 5 nodes. We used the same protocol as followed in [23]. We used the Curve, Hue/Saturation, Channel Mixer, Brightness, Vibrance adjustment options and blur filters for generating the test set consisting of Photoshopped images.

We performed another experiment where we wanted to test the generalizability of the proposed method in terms of **unseen modalities** and **unseen configurations**. For this evaluation, we tested on 6,000 near-infrared iris images from the CASIA Iris V2 Device 2 dataset [6] corresponding to 30 subjects, resulting in 1,200 IPTs, where each IPT contains 5 images (see Figure 7.7(a)).

Finally, we conducted an experiment to evaluate whether the GCN-based node embedding module can handle **unseen number of nodes**. The GCN is trained using IPT configurations comprising 5 nodes, but we tested it on a publicly available Near-duplicate Face Images (NDFI) –Set I [21] dataset comprising 1,229 IPTs, where each IPT consists of 10 nodes. Therefore, the test set for this experiment comprises 12,290 images.

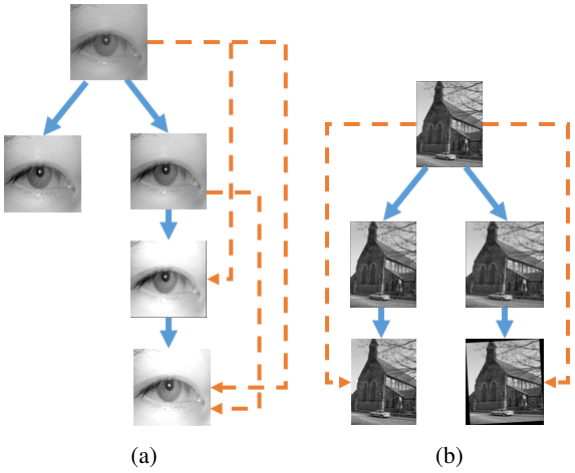


Figure 7.7: IPT configuration of iris and natural scene images used for evaluation in Experiment 2. The configuration used in iris near-duplicates is different from the ones used in training (see Figure 7.6). The immediate links are depicted using bold blue arrows, while the ancestral links are depicted using dashed orange arrows.

7.4.3 Experiment 3: Evaluation of the proposed IPF reconstruction using locally-scaled spectral clustering and GCN-based node embedding and PRNU-based link prediction

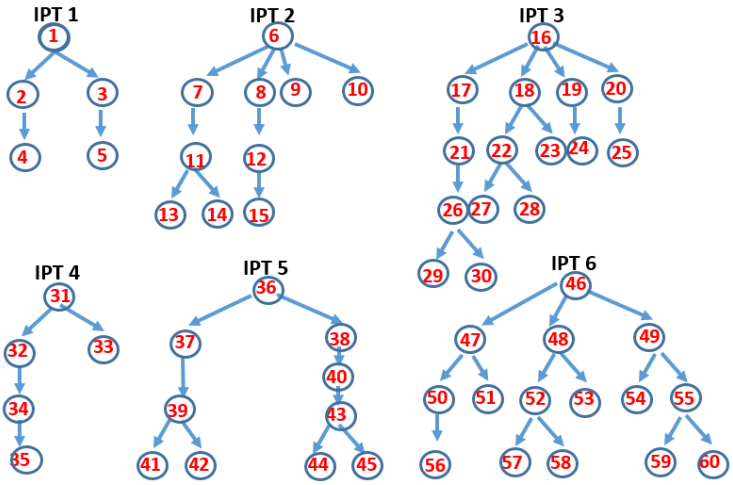


Figure 7.8: Illustration of the image phylogeny forest structures used in Experiment 3. Each IPF comprises three IPTs, where each IPT may have 5 nodes (IPT 1 and IPT 4) or 10 nodes (IPT 2 and IPT 5) or 15 nodes (IPT 3 and IPT 6). The selected test configurations differ from the configurations used in training the GCN and indicate variations both in density and configurations of the IPF test cases. The immediate links are indicated for ease of visualization but ancestral links are also included for evaluation.

For the IPF reconstruction we used face images from the WVU Multimodal Release I dataset [12]. We used only those subjects whose images have been acquired using two different sensors: Sony EVI- D30 and Sony EVI- D39. Therefore, we used images from 49 subjects. We randomly selected 3 sample images from each of the 49 subjects. Next we subjected the images to seven transformations (four photometric and three geometric as described in Table 7.1). We used the six configurations depicted in Figure 7.8 to generate 294 near-duplicate sets (49 subjects $\times$ 3 IPT configurations in each IPF $\times$ 2 sensors), such that each IPF will contain three IPTs in total - the first IPT with 5 nodes, the second IPT with 10 nodes and the third IPT with 15 nodes, resulting in a total of 2,940 images (49 IPFs $\times$ 30 images in each IPF). Examples of some near-duplicates used in IPF reconstruction are presented in Figure 7.1. The test set contains variations in **sensors**, **subjects** (11 females and 38 males) and **acquisition settings** (indoor and outdoor). We used three IPFs to compute the parameter for locally-scaled spectral clustering (η = 0.7, see Algorithm 7). We report the results firstly, in terms of the performance of the locally-scaled spectral clustering

algorithm (mean number of clusters and clustering accuracy) and then secondly, in terms of root identification and IPF reconstruction accuracies.

7.4.4 Baseline

We compare the proposed locally-scaled clustering with conventional spectral clustering algorithm. We compare the proposed GCN and PRNU based IPT reconstruction with Gaussian RBF and Chebyshev polynomials basis functions approach [23], and the Oriented Kruskal algorithm [61,62]. The codes for implementing autoencoder-based method [38] and transformation aware embedding-based method [32] are not open source, and therefore, could not be used as baselines for comparison.

7.5 Results and Analysis

7.5.1 Results for Experiment 1

The proposed locally-scaled spectral clustering is evaluated for near-duplicates downloaded from the Internet and images generated using deep learning scheme. Figure 7.9(a) indicates that the images are clustered into six distinct IPTs. Although there is no ground truth associated with the images for empirical evaluation, we can do a visual inspection to perform a qualitative analysis in this case. Some images seem to have been modified by inserting digital watermarking (such as '3' and '4') and are correctly assigned to separate clusters. Three images have been assigned to IPT '5', although visual inspection reveals that two out of three images are correctly assigned, while the third image should haave been assigned to a separate cluster. The results in Figure 7.9(b) appear to be correct with two clusters.

For the near-duplicates generated using deep learning-based method, each image is generated by modifying attributes such as adding hair bangs or adding glasses. Further modifications include making the shades darker or changing the direction of the hair bangs or making them dense or sparse. The modifications are performed mostly on the original image, so disjoint IPTs are anticipated as also indicated by the locally-scaled clustering outputs in a majority of cases.

(a) (b)

Figure 7.9: Experiment 1: Locally-scaled spectral clustering performance for near-duplicates downloaded from the Internet. The numbers indicate the cluster identifier to which an image has been assigned. On the left, six clusters (IPTs) have been identified. On the right, two clusters (IPTs) have been identified. The results are for visual inspection only as no ground truth is associated with them.

7.5.2 Results for Experiment 2

We first compared the performance of the three node embedding techniques and observed that GCN (with Chebyshev polynomial of degree 3) outperforms GCN-Linear and HGNN by a considerable margin ($\approx 20\%$), and is, therefore, selected as the best node embedding technique. We subsequently used the depth labels provided by the GCN technique for IPT reconstruction in the link prediction module. The reason for GCN to perform better than the remaining two methods could be attributed to the use of a higher degree polynomial compared to GCN-Linear and HGNN (both use Chebyshev polynomial of degree 1). The spectral convolution filter is approximated using a truncated Chebyshev polynomial expansion. We hypothesize that higher order polynomials can therefore perform effective spectral convolutions, resulting in accurate node embedding. In contrast, the linearization used in GCN-Linear and HGNN allows deeper architecture and combination of multiple features, but weakens the ability to model global relationships. Thus, GCN outperforms both GCN-Linear and HGNN, and is used for evaluating the remaining experiments.

Next, we evaluated the performance of the GCN-based node embedding and PRNU-based link prediction module in terms of root identification and IPT reconstruction accuracies in the two

Figure 7.10: Experiment 1: Locally-scaled spectral clustering performance for near-duplicates generated using deep learning-based transformations [84]. The numbers indicate the cluster identifier to which an image has been assigned. The proposed method can successfully discern between minute changes in the attributes and assigns the modified images to distinct clusters (IPTs) in a majority of cases.

scenarios. In Scenario 1, where both training and testing is conducted on face images, the proposed method achieves a root identification accuracy of 85.11% in the context of photometrically modified images and 73.22% in the context of geometrically modified images, averaged across the six configurations. In terms of IPT reconstruction accuracies, the proposed method achieves 90.64% in the context of photometrically modified images and 87.31% in the context of geometrically modified images, averaged across the six configurations. In Scenario 2, where training is performed using face images but testing is conducted on images containing natural scene, the proposed method achieves a root identification accuracy of 74.67% in the context of photometrically modified images and 75.33% in the context of geometrically modified images, averaged across the six configurations. In terms of IPT reconstruction accuracies, the proposed method achieves 87.60% in the context of photometrically modified images and 86.72% in the context of geometrically modified images, averaged across the six configurations. See Table 7.2. The results indicate that some configurations particularly II and VI are very difficult to reconstruct both for photometric

Table 7.2: Experiment 2: Performance of node embedding and link prediction modules in terms of root identification and IPT reconstruction accuracies for both photometric and geometric transformations. The results are reported for two scenarios. The values to the left of the forward slash indicate Scenario 1 (trained on face images and tested on face images) and the values to the right indicate Scenario 2 (trained on face images but tested on images depicting natural scenes).

IPT configuration	Photometric transformations		Geometric transformations	
	Root identification accuracy (%)	IPT reconstruction accuracy (%)	Root identification accuracy (%)	IPT reconstruction accuracy (%)
IPT A	90.0 / 88.0	94.0 / 91.33	86.67 / 90.0	91.67 / 87.33
IPT B	69.33 / 48.0	78.80 / 74.80	46.67 / 58.0	74.47 / 76.0
IPT C	86.0 / 76.0	96.0 / 89.67	74.67 / 72.0	93.78 / 88.0
IPT D	98.67 / 98.0	97.17 / 100.0	95.33 / 96.0	95.33 / 100
IPT E	94.0 / 92.0	97.73 / 93.60	93.33 / 86.0	96.27 / 94.0
IPT F	72.67 / 46.0	80.15 / 76.22	42.67 / 50.0	72.37 / 74.22
Average	85.11 / 74.67	90.64 / 87.60	73.22 / 75.33	87.31 / 86.72

and geometric transformations, indicating the difficulty in reconstructing deeper and unbalanced trees. The results indicate that the proposed GCN-based node embedding and PRNU-based link prediction modules are adept in handling not only different classes of transformations (photometric and geometric), but also different types of images (biometric and generic images).

In the context of *unseen transformations*, the proposed method acheieves a root identification accuracy of 82.86% and an IPT reconstruction accuracy of 92.95% averaged across four IPT configurations (see Table 7.3). In the context of *unseen modalities and configurations*, the proposed method achieves a root identification accuracy of 85.92% and an IPT reconstruction accuracy of 90.85% in the case of iris images. In the context of *unseen number of nodes*, the proposed method achieves a root identification accuracy of 90.97%, and an IPT reconstruction accuracy of 61.02%. The best performing method in [17] reported a root identification accuracy of 89.91% at Rank 3, and an IPT reconstruction accuracy of 70.61%, assuming that the the root node is known apriori. In contrast, we report only one root identification accuracy and we make no assumptions about using the correct root node for IPT reconstruction. Overall, the results indicate that the proposed method is capable of handling unseen transformations, modalities, configurations and number of nodes reliably well.

Table 7.3: Experiment 2: Evaluation of the performance of node embedding and link prediction modules in the context of unseen transformations, unseen modalities and configurations and unseen number of nodes.

Experimental settings	Root identification accuracy (%)	IPT reconstruction accuracy (%)
Unseen transformations	82.86	92.95
Unseen modalities and configurations	85.92	90.85
Unseen number of nodes	90.97	61.02

7.5.3 Results for Experiment 3

In terms of IPF reconstruction, we evaluate both proposed methods: locally-scaled spectral clustering for identifying number of IPTs and then node embedding and link prediction module for IPF reconstruction (construct each IPT within the IPF). Table 7.4 reports the number of clusters produced by conventional spectral clustering and the proposed locally-scaled spectral clustering. Figure 7.11 indicates the clustering accuracies (*i.e.* the proportion of images correctly assigned to the respective clusters) for both conventional and proposed spectral clustering methods. Each IPF contains three clusters (IPTs), and each IPT comprises either 5 or 10 or 15 nodes. Results indicate that conventional clustering produces much higher clusters than the desired output and result in erroneous assignment of images. On the other hand, the proposed method performs well irrespective of the number of nodes. The global bandwidth used in conventional spectral clustering has been computed using the standard deviation of the respective features (face descriptors, PRNU and pixel intensities). A single global bandwidth is not suited to correctly identify the number of IPTs as evident from the findings, and substantiates the importance of locally-scaled spectral clustering.

In terms of root identification and IPF reconstruction accuracies, we report them separately for each IPT configuration. We reconstruct the IPT using Oriented Kruskal, Gaussian RBF (Gaussian RBF outperformed Chebyshev polynomials, so we are reporting the results pertaining to Gaussian RBF only for the sake of brevity) and the proposed method from the clustered outputs of locally-scaled clustering algorithm. In all cases, the proposed GCN-based node embedding and PRNU-

Table 7.4: Experiment 3: Number of clusters (mean and standard deviation) produced during IPF construction by conventional spectral clustering and locally-scaled spectral clustering (proposed). A lower value (mean ≈ 1, standard deviation ≈ 0) is desirable. The proposed method yields better results (bolded).

Number of nodes	Number of clusters produced during IPF construction (mean ± std. deviation)	
	Spectral clustering	Locally-scaled spectral clustering (Proposed)
5	2.31 ± 1.04	**1.70 ± 0.54**
10	1.98 ± 0.21	**1.52 ± 0.32**
15	2.35 ± 0.80	**1.84 ± 0.53**

Table 7.5: Experiment 3: Evaluation of GCN-based node embedding and PRNU-based link prediction for each IPT configuration used in the IPF in terms of root identification and reconstruction accuracies. Results indicate that the proposed method (bolded) significantly outperforms state-of-the-art baselines in all the cases.

IPT configuration	Root identification accuracy (%)			IPT reconstruction accuracy (%)		
	Oriented Kruskal	Basis functions (Gaussian RBF)	GCN+PRNU	Oriented Kruskal	Basis functions (Gaussian RBF)	GCN+PRNU
IPT 1 (5 nodes)	32.61	23.91	**78.26**	21.56	34.17	**66.78**
IPT 2 (10 nodes)	7.24	12.32	**57.97**	15.38	28.09	**71.48**
IPT 3 (15 nodes)	13.04	7.24	**47.10**	14.96	24.01	**61.55**
IPT 4 (5 nodes)	27.53	42.03	**79.71**	19.02	40.37	**60.76**
IPT 5 (10 nodes)	21.74	35.50	**55.79**	16.59	31.09	**62.03**
IPT 6 (15 nodes)	11.59	17.39	**31.88**	15.43	31.11	**58.86**
Average	18.96	23.06	**46.97**	17.16	31.47	**59.41**

based link prediction outperform the two baselines employing Oriented Kruskal and basis functions (see Table 7.5) by a significant margin. We have also reported the average root identification and IPT reconstruction accuracies, which measures the overall IPF reconstruction performance. Results indicate that the proposed method outperforms Oriented Kruskal by 28.01% and Gaussian RBF by 23.91% in terms of root identification accuracy; the proposed method outperforms Oriented Kruskal by 42.25% and Gaussian RBF by 27.94% in terms of IPT reconstruction accuracy. Error plots indicating mean and standard deviations of the root identification and reconstruction accuracies are illustrated as a function of the variation in the number of nodes (images) in Figure 7.12. Results indicate the proposed method outperforms Gaussian RBF, which in turn outperforms Oriented Kruskal method. The results also indicate that GCN which has been trained on five images *only* can work fairly well on any *arbitrary* number of images, such as ten or fifteen.

The **main findings** from the experiments are as follows:

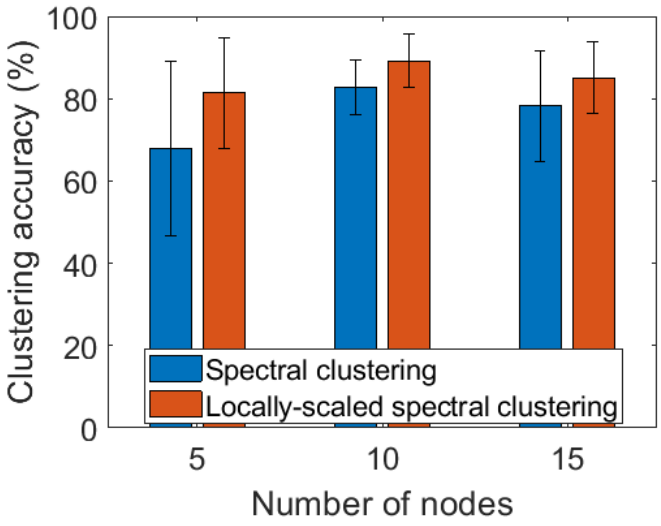


Figure 7.11: Experiment 3: Variation of clustering accuracies as a function of the number of nodes for the conventional spectral clustering (blue) and the locally-scaled spectral clustering (proposed) methods. The proposed method (orange bars) consistently results in higher means and lower standard deviations in clustering accuracies across 5, 10 and 15 nodes over the conventional spectral clustering algorithm.

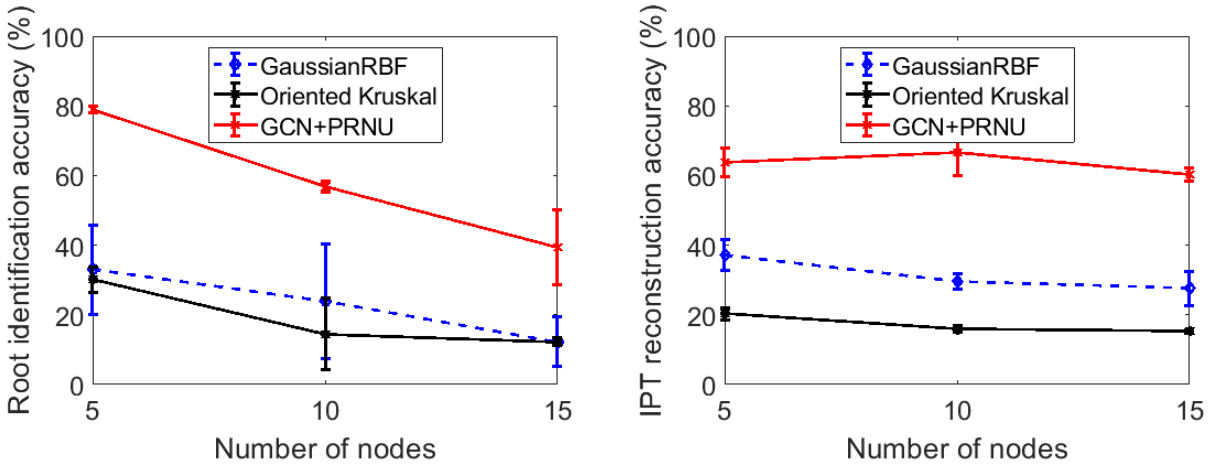


Figure 7.12: Experiment 3: Variation in root identification and IPT reconstruction accuracies as a function of the number of nodes.

1. Locally-scaled spectral clustering can suitably address the multi-scale issue encountered by conventional spectral clustering. This is particularly relevant in IPF reconstruction where, the number of nodes (near-duplicate images) belonging to each IPT is not known *apriori* (see Table 7.4). Experiments indicate that the proposed method performs robustly when the

number of nodes in an IPT is varied by a factor of two and three in an IPF, and appropriates well to near-duplicates downloaded from the Internet or generated using deep learning tools.

2. Graph convolutional network-based node embedding and PRNU-based link prediction can significantly outperform existing methods in constructing image phylogeny tree. Experiments indicate that the proposed method performs very well for both photometrically and geometrically modified face as well as natural scene images (see Table 7.2). The proposed method also generalizes well across unseen transformations, unseen modalities, unseen configurations and unseen number of nodes. See Table 7.3.

3. Locally-scaled spectral clustering used in conjunction with GCN and PRNU can successfully outperform existing methods in the context of image phylogeny forest reconstructions and offers substantial improvement (upto $\approx 28\%$ in terms of root identification and upto $\approx 42\%$ in terms of IPF reconstruction accuracies, see Table 7.5).

7.6 Summary

In this chapter, we explored the use of graph convolutional network for the purpose of image phylogeny. Firstly, we deduce the depth label at which an image is likely to be present in the image phylogeny tree. Secondly, we used sensor pattern noise features (PRNU) extracted from the images to predict links between parent and child nodes to finally construct the IPT. By leveraging graph-based approach and PRNU-based analysis, we judiciously combined global and local analyses in constructing IPTs with higher reconstruction accuracies than existing methods, which mostly focus on pairwise analysis. Furthermore, the method generalized well across unseen transformations, unseen configurations and across modalities. We successfully applied this method to construct image phylogeny forests (comprising IPTs of arbitrary number of nodes and structures) by using a locally-scaled spectral clustering technique. The resultant approach outperformed state-of-the-art methods by upto $\approx 28\%$ in terms of root identification accuracy and upto $\approx 42\%$ in terms of IPF reconstruction accuracy.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Research Contributions

Research in the context of digital image forensics has existed over several decades. But the same cannot be ascertained for biometric images. The widespread utility of biometrics in a myriad of applications combined with the availability of inexpensive image editing tools, has therefore, necessitated the importance of examining forensics for digitally modified biometric images. It will undoubtedly prove to be useful in commercial and judicial services.

In this thesis, we explore digitally edited biometric images from two perspectives –(i) *sensor-based analysis* and (ii) *content-based analysis*. The sensor-based analysis encompasses the sensor (camera) details of an image, as it can provide useful cues about the processing history of an image. Sensor details present in an image can be used for copyright protection and deducing what sensor has been used to acquire an image. This is particularly relevant in biometrics, as the sensors used in acquiring biometric images vary across modalities. For example, fingerprint images are acquired using capacitive or optical sensors, iris images are acquired using near-infrared sensors, and face images are acquired using sensors operating in the visible spectrum. Images acquired using different sensors will leave unique traces in the images, which can be extracted for sensor identification routines. On the other hand, the content-based analysis covers modifications applied to the content present in the image. These changes can be very subtle, such as photometric and geometric transformations which may be visually imperceptible and result in near-duplicates. In the case of biometric images, such transformations may interfere with their matching utility. For example, face images from surveillance videos can be contrast adjusted to increase the matching performance of a commercial matcher, which will violate the chain of custody of digital evidence. Also, the image characteristics vary across different modalities. For example, face images contain structural details while fingerprint images contain textural details. So the content-based analysis of

near-duplicates can help deduce the original image as well as determine the trail of modifications. Therefore, we pursued both degrees of analysis, and leveraged them in a unified framework for improved forensic analysis of biometric images.

In the context of biometric sensor identification, we analyzed the feasibility of Photo Repsonse Non-Uniformity (PRNU) in correctly deducing the near-infrared sensor used to acquire ocular images. PRNU has been used for sensor identification of generic images, but limited work has been done in the context of application of PRNU for biometric images. We further examined the impact of photometric transformations on the reliability of PRNU-based iris sensor identification. The variation in illumination conditions can induce photometric transformations in biometric images. Our findings substantiated that certain illumination normalization schemes can adversely impact iris sensor recognition.

To further test the robustness of the PRNU-based sensor identification schemes, we also developed methods to deliberately confound sensor recognition methods. Another objective of sensor de-identification is to cater to privacy preservation. Sensor recognition can be implicitly linked to the individual possessing the device. By removing the sensor-specific traces while maintaining the image content, one can unlink the device and the user. In our case of biometric images, we retained their biometric utility while impeding their sensor identifiability. In the first method, we applied perturbations to patches of ocular images and repeated this process until the image is misclassified as belonging to a different sensor than its original acquisition device. This iterative process was able to confound a specific PRNU-based classifier. In the second method, we developed a sensor de-identification technique that used Discrete Cosine Transform to suppress sensor traces in a single pass, and generalized to multiple PRNU classifiers.

Furthermore, we developed a method that used deep learning to jointly learn biometric and sensor details present in an image using a one-shot approach. The joint representation can be used for the task of performing biometric and sensor recognition simultaneously. The task of joint biometric-device recognition can be used for authentication on current smartphones that use biometric signature of the owner for access control, such as FaceID on Apple's iPhones. The

joint representation couples biometric and sensor signatures non-trivially, and therefore, implicitly imparts privacy to the biometric template.

Next, we probed into the *content-based* analysis of the images, particularly, in the task of image phylogeny for near-duplicate biometric images. Photometric and geometric transformations can be digitally induced, repeatedly, resulting in numerous near-duplicates that are manually indiscernible from the original image. However, identifying the original image and understanding the evolution of the near-duplicates is important from the perspective of media forensics. The task of image phylogeny is highly challenging as the scope of transformations and the widespread distribution of edited content on online platforms keep evolving. To the best of our knowledge, we initiated the research involving image phylogeny for biometric images. We developed two methods to combat this difficult problem. In the first method, we used a deterministic approach to model transformations between a pair of near-duplicate images and used the estimated parameters to construct the image phylogeny tree (IPT). The method did not consider any prior information while inferring directed relationships between the set of near-duplicates. So in the second method, we leveraged a probabilistic approach, where we used the likelihood ratio of the estimated parameters computed from a training set to deduce the original image and the hierarchical structure depicting the IPT. We further evaluated the proposed method using different families of basis functions, and observed that the Chebyshev polynomials and Gaussian radial basis functions were the best candidates for inferring the IPT.

Both deterministic and probabilistic approaches for IPT (re)construction involved pairwise modeling which disregarded the global information. Therefore, in our final work we utilized a graph-based approach, specifically a graph convolutional network to inspect the global relationships between a set of near-duplicates by examining all the images simultaneously. In this work, we also utilized the sensor pattern noise (PRNU) features to distinguish between original and transformed images, which assisted in correctly identifying the parent and the child node in the IPT. The motivation behind using PRNU was driven by existing works that indicated the variation in the PRNU in response to geometric transformations, and our own examinations of the impact of

photometric transformations on PRNU. The proposed method combined sensor and content-based analyses to tackle the task of image phylogeny for near-duplicate face images. It robustly handled unseen transformations, different biometric modalities and arbitrary number of images within the IPT. It demonstrated promising results when tested on image phylogeny forests comprising multiple IPTs with different structures. The proposed method outperformed existing state-of-the-art methods by a significant margin.

As far as computational costs are concerned, for a 10-node Image Phylogeny Tree:
(i) Basis functions-based framework needs 12.98 secs. (12.96 secs. for dissimilarity matrix computation; 0.02 secs. for root node identification and IPT reconstruction)
(ii) GCN and Sensor pattern noise needs 4.45 secs. (4.37 secs. for depth labels; 0.08 secs. for link prediction)
All evaluations are done using Intel® Core ™ i7-7700 CPU @ 3.6 GHz.

As far as limitations are concerned, our analysis reveals that the performance of the method is limited when the variation in the images between different depth labels does not possess a continuous gradient. For example, if the images located at higher depth labels are very similar to the images located at lower depth labels (indicative of a cycle likely to be present), then the modeling of transformations using basis functions as well as the inference of the depth labels by the Graph Convolutional Network tend to be erroneous. This error propagates if the original structure is depth-heavy, and is worst for trees with no sibling nodes (no breadth). In our test cases, we randomly simulated the transformations without enforcing any gradient constraint on successive transformations. As a result, we observed poor performance for trees with deep configurations. However, in cases where there is gradual increase of transformations, our method was able to correctly identify the root node and fully recover the phylogeny structure, as indicated in the example (see Figure 6.6(b)) involving near-duplicates generated by gradually increasing the intensity of the makeup.

8.2 Future Work

Image manipulation has been augmented and has achieved a new level of "realism" through the courtesy of deep learning. DeepFakes have emerged and have piqued the interest of academicians, researchers, engineers, and the government. Although it made its first appearance as a feat achieved by deep learning networks, quickly enough, the prospect of its abuse for misinformation and disinformation has raised concerns. Distinguishing between 'real' and 'fake' images has therefore become of paramount importance. We will conclude this thesis by touching on some future directions by extending the current works.

1. In the thesis, we focused primarily on the task of image phylogeny. The same principle can be applied to the task of video phylogeny. Only one work has been done in the context of video phylogeny, in which the temporal sequence of the video frames is depicted using the phylogenetic structure. The idea is to screen near-duplicate frames and use a reduced subset of frames, followed by application of existing image phylogeny tree construction routines to deduce the hierarchical structure. The task is highly challenging due to the variation in the scene and time, simultaneously, but will be extremely useful in reconstructing timeline for crime scenes. This will be particularly relevant in public incidents, such as riots or bombings, where potentially, several people might capture the scene using their smartphone cameras.

2. The task of image phylogeny requires discriminating between original and transformed images. The integration of graph-based approach and sensor pattern noise can be used as a stepping stone towards identifying DeepFakes. Current DeepFake detection methods use DeepFake images during training, and can therefore, accurately detect DeepFakes generated using a particular algorithm. However, the generalizability of the current methods to new DeepFake generators seem to be still lacking. In contrast, the graph-based approach used in image phylogeny, models the underlying manifold of the data, and can be used to disentangle between the subspaces populated by the real and the fake images. Also, synthetic images will

not carry sensor traces typically, so the use of sensor pattern noise may additionally assist in discriminating between the two subspaces.

3. The use of graph convolutional network and sensor pattern noise can be leveraged to detect morphed images or even biometric presentation attacks. One of the advantages of the graph-based approach was its generalizability to unseen transformations and unseen biometric modalities. Morphing typically involves fusing two images, and can be applied to face images and iris images. Morphing of biometric images aggravates the concern of falsely matching the morphed image to two separate identities. Presentation attacks, on the other hand, circumvent the biometric recognition system by presenting an altered biometric sample, which can be in the form of replay attacks, print attacks, plastic eyes or face masks. Distinguishing between bonafide face images and presentation attacks or morphed images is a pressing problem that may be addressed using the graph convolutional network.

4. Cyberattacks such as website defacements can use disturbing graphics to incite unrest among the audience. The hacker may reuse these images with subtle modifications as an insignia of their propaganda. Tracking these near-duplicates through the proposed approach can deliver some useful insight into the modus operandi of the defacer. This will help track perpetrators and can assist the authorities in boosting cybersecurity.

5. Authentication using biometric signatures will become ubiquitous in the futuristic smart city. Smart home environments already use face recognition to monitor guests arriving at the entrance, smart mobility uses ECG-based signatures to solve drowsy driver problem. In all these cases, the seamless authentication of the device and the user is essential. The joint representation that we developed for performing smartphone sensor and biometric recognition simultaneously, can be extended to work across different platforms. Although our work focused primarily on the performance of the joint representation, future work can delve into the computational efficiency, memory requirement and template protection strategies for the combined biometric-device representation.

We will also consider multiple sources for a single image as part of our Future Work. This task will come under the scope of 'provenance analysis' [124] which first identifies the relevant donor images for a single composite image using a provenance filtering step, and then follow it with a provenance graph construction step for determining the order of modifications. We will also consider locating editing boundaries in images which have been locally tampered with. This task will come under the scope of 'manipulated image detection'.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Iris.htm.

[2] http://www.iab-rubric.org/resources/impdatabase.html.

[3] https://sites.google.com/a/nd.edu/public-cvrl/data-sets.

[4] http://www.cs.princeton.edu/~andyz/irisrecognition.

[5] Augmentor: Image augmentation library in python for machine learning. https://github.com/mdbloice/Augmentor. [Online accessed: 3rd Januray, 2020].

[6] CASIA Iris Database Version 2. http://biometrics.idealtest.org/dbDetailForUser.do?id=2. [Online accessed: 12th April 2019].

[7] CASIA Iris Database Version 4. http://biometrics.idealtest.org/dbDetailForUser.do?id=4. [Online accessed: 30th August 2019].

[8] CelebA dataset. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html. [Online accessed: 3rd January, 2020].

[9] Embedding network tutorial. https://github.com/adambielski/siamese-triplet/blob/master/Experiments_MNIST.ipynb. [Online accessed: 12th January, 2020].

[10] Smartphone pictures poses privacy risks. https://www.youtube.com/watch?v=N2vARzvWxwY&feature=youtu.be. [Online accessed: 18th April 2019].

[11] VeriEye iris matcher. https://www.fulcrumbiometrics.com/Iris-Matcher-License-p/100424.htm. [Online: accessed 13-December-2018].

[12] WVU multimodal dataset release 1. https://biic.wvu.edu/data-sets/multimodal-dataset. [Online accessed: 15th March, 2020].

[13] A. Agarwal, A. Sehwag, R. Singh, and M. Vatsa. Deceiving face presentation attack detection via image transforms. *IEEE International Conference on Multimedia Big Data*, 08 2019.

[14] Akshay Agarwal, Rohit Keshari, Manya Wadhwa, Mansi Vijh, Chandani Parmar, Richa Singh, and Mayank Vatsa. Iris sensor identification in multi-camera environment. *Information Fusion*, 45:333 – 345, 2019.

[15] R. Arjona, M. A. Prada-Delgado, I. Baturone, and A. Ross. Securing minutia cylinder codes for fingerprints through physically unclonable functions: An exploratory study. In *Proc. of 11th IAPR International Conference on Biometrics (ICB)*, Gold Coast, Australia, June 2018.

[16] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[17] S. Banerjee, V. Mirjalili, and A. Ross. Spoofing PRNU patterns of iris sensors while preserving iris recognition. In *5th IEEE International Conference on Identity, Security, Behavior and Analysis (ISBA)*, January 2019.

[18] S. Banerjee and A. Ross. From image to sensor: Comparative evaluation of multiple PRNU estimation schemes for identifying sensors from NIR iris images. In *Fifth International Workshop on Biometrics and Forensics*, 2017.

[19] S. Banerjee and A. Ross. Impact of photometric transformations on PRNU estimation schemes: A case study using near infrared ocular images. In *International Workshop on Biometrics and Forensics (IWBF)*, pages 1–8, June 2018.

[20] S. Banerjee and A. Ross. Face phylogeny tree: Deducing relationships between near-duplicate face images using legendre polynomials and radial basis functions. In *10th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Tampa, Florida*, September 2019.

[21] S. Banerjee and A. Ross. Face phylogeny tree: Deducing relationships between near-duplicate face images using legendre polynomials and radial basis functions. In *10th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, September 2019.

[22] S. Banerjee and A. Ross. Smartphone camera de-identification while preserving biometric utility. In *Proc. of 10th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Tampa, USA, September 2019.

[23] S. Banerjee and A. Ross. Face phylogeny tree using basis functions. In *IEEE Transactions on Biometrics, Behavior and Identity Science*, 2020.

[24] N. Bartlow, N. Kalka, B. Cukic, and A. Ross. Identifying sensors from fingerprint images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 78–84, June 2009.

[25] N. Bartlow, N. Kalka, B. Cukic, and A. Ross. Identifying sensors from fingerprint images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 78–84, 2009.

[26] A. Bartoli. Groupwise geometric and photometric direct image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2098–2108, Dec 2008.

[27] S. Bayram, H. Sencar, N. Memon, and I. Avcibas. Source camera identification based on CFA interpolation. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–69–72, Sept 2005.

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

[29] T. Berry and T. Sauer. Spectral clustering from geometric viewpoint. Technical report, 2015.

[30] P. Bestagini, M. Tagliasacchi, and S. Tubaro. Image phylogeny tree reconstruction based on region selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2059–2063, March 2016.

[31] A. Bharati, D. Moreira, A. Pinto, J. Brogan, K. Bowyer, P. Flynn, W. Scheirer, and A. Rocha. U-phylogeny: Undirected provenance graph construction in the wild. In *IEEE International Conference on Image Processing (ICIP)*, 05 2017.

[32] Aparna Bharati, Daniel Moreira, Patrick J. Flynn, Anderson Rocha, Kevin W. Bowyer, and Walter J. Scheirer. Learning transformation-aware embeddings for image forensics. *ArXiv*, abs/2001.04547, 2020.

[33] G. Bhupendra and M. Tiwari. Improving source camera identification performance using DCT based image frequency components dependent sensor pattern noise extraction method. *Digital Investigation*, 03 2018.

[34] D. Bobeldyk and A. Ross. Analyzing covariate influence on gender and race prediction from near-infrared ocular images. pages 7905–7919, 2019.

[35] Z. Boulkenafet, J. Komulainen, Lei. Li, X. Feng, and A. Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2017.

[36] Z. Boulkenafet, J. Komulainen, Lei. Li, X. Feng, and A. Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2017.

[37] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proc. of 6th International Conference on Neural Information Processing Systems*, pages 737–744, 1993.

[38] R. Castelletto, S. Milani, and P. Bestagini. Phylogenetic minimum spanning tree reconstruction using autoencoders. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2817–2821, 2020.

[39] Hung-Jen Chen, Ka-Ming Hui, Szu-Yu Wang, Li-Wu Tsao, Hong-Han Shuai, and Wen-Huang Cheng. BeautyGlow: On-demand makeup transfer framework with reversible generative network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] L. Chen, L. Xu, X. Yuan, and N. Shashidhar. Digital forensics in social networks and the cloud: Process, approaches, methods, tools, and challenges. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 1132–1136, Feb 2015.

[41] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, March 2008.

[42] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, March 2008.

[43] W. Chen, M. J. Er, and S. Wu. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(2):458–466, April 2006.

[44] Sumit Chopra, Raia Hadsell, and Yann Lecun. Learning a similarity metric discriminatively, with application to face verification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:539– 546, 07 2005.

[45] J. Ćosić and M. Bača. (Im)proving chain of custody and digital evidence integrity with time stamp. In *The 33rd International Convention MIPRO*, pages 1226–1230, May 2010.

[46] Jasmin Ćosić and Zoran Ćosić. Chain of custody and life cycle of digital evidence. In *Journal of Computer Technology and Applications*, volume 3, pages 126–129, February 2012.

[47] F. Costa, A. Oliveira, P. Ferrara, Z. Dias, S. Goldenstein, and A. Rocha. New dissimilarity measures for image phylogeny reconstruction. *Pattern Analysis and Applications*, 20, 03 2017.

[48] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, Jan 2004.

[49] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. In *Journal of Optical Society of America A*, volume 2, pages 1160–1169, July 1985.

[50] F. de O. Costa, M. A. Oikawa, Z. Dias, S. Goldenstein, and A. R. de Rocha. Image phylogeny forests reconstruction. *IEEE Transactions on Information Forensics and Security*, 9(10):1533–1546, Oct 2014.

[51] A. A. de Oliveira, P. Ferrara, A. De Rosa, A. Piva, M. Barni, S. Goldenstein, Z. Dias, and A. Rocha. Multiple parenting phylogeny relationships in digital images. *IEEE Transactions on Information Forensics and Security*, 11(2):328–343, 2016.

[52] L. Debiasi and A. Uhl. Blind biometric source sensor recognition using advanced PRNU fingerprints. In *23rd European Signal Processing Conference (EUSIPCO)*, pages 779–783, Aug 2015.

[53] L. Debiasi and A. Uhl. Techniques for a forensic analysis of the CASIA-IRIS V4 database. In *3rd International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, March 2015.

[54] L. Debiasi and A. Uhl. Comparison of PRNU enhancement techniques to generate PRNU fingerprints for biometric source sensor attribution. In *4th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, March 2016.

[55] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3844–3852, 2016.

[56] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[57] Z. Dias, S. Goldenstein, and A. Rocha. Exploring heuristic and optimum branching algorithms for image phylogeny. *Journal of Visual Communication and Image Representation*, 24(7):1124 – 1134, 2013.

[58] Z. Dias, S. Goldenstein, and A. Rocha. Large-scale image phylogeny: Tracing image ancestral relationships. *IEEE MultiMedia*, 20(3):58–70, July 2013.

[59] Z. Dias, A. Rocha, and S. Goldenstein. First steps toward image phylogeny. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, Dec 2010.

[60] Z. Dias, A. Rocha, and S. Goldenstein. First steps toward image phylogeny. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, Dec 2010.

[61] Z. Dias, A. Rocha, and S. Goldenstein. Image phylogeny by minimal spanning trees. *IEEE Transactions on Information Forensics and Security*, 7(2):774–788, April 2012.

[62] Zanoni Dias, Siome Goldenstein, and Anderson Rocha. Toward image phylogeny forests: Automatically recovering semantically similar image relationships. *Forensic Science International*, 231(1–3):178 – 189, 2013.

[63] A. E. Dirik and A. Karaküçük. Forensic use of photo response non-uniformity of imaging sensors and a counter method. *Opt. Express*, 22(1):470–482, Jan 2014.

[64] 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*, 9(12):2277–2290, Dec 2014.

[65] H. Farid. Digital image forensics. *Scientific American*, 298(6):66–71, 2008.

[66] H. Farid. *Photo Forensics*. The MIT Press, 2016.

[67] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[68] J. J. Foo, J. Zobel, and R. Sinha. Clustering near-duplicate images in large collections. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, page 21–30, New York, NY, USA, 2007. Association for Computing Machinery.

[69] David Freire-Obregón, Fabio Narducci, Silvio Barra, and Modesto Castrillón-Santana. Deep learning for source camera identification on mobile devices. *Pattern Recognition Letters*, 126:86 – 91, 2019. Robustness, Security and Regulation Aspects in Current Biometric Systems.

[70] J. Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, March 2009.

[71] C. Galdi, M. Nappi, and J. L. Dugelay. Multimodal authentication on smartphones: Combining iris and sensor recognition for a double check of user identity. *Pattern Recognition Letters*, 3:34–40, 2015.

[72] Chiara Galdi, Michele Nappi, and Jean-Luc Dugelay. Combining hardwaremetry and biometry for human authentication via smartphones. In Vittorio Murino and Enrico Puppo, editors, *Image Analysis and Processing (ICIAP)*, pages 406–416. Springer International Publishing, 2015.

[73] Chiara Galdi, Michele Nappi, and Jean-Luc Dugelay. Secure user authentication on smartphones via sensor and face recognition on short video clips. In Man Ho Allen Au, Arcangelo Castiglione, Kim-Kwang Raymond Choo, Francesco Palmieri, and Kuan-Ching Li, editors, *Green, Pervasive, and Cloud Computing*, pages 15–22. Springer International Publishing, 2017.

[74] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, and K. Kuroki. Methods for identification of images acquired with digital cameras. *Proc. SPIE 4232, Enabling Technologies for Law Enforcement and Security*, 2001.

[75] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme. Can we trust digital image forensics? In *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 78–86. 01 2007.

[76] M. Goljan. Digital camera identification from images - Estimating false acceptance probability. *Proc. 7th International Workshop on Digital Watermarking*, Nov 2008.

[77] M. Goljan, J. Fridrich, and M. Chen. Sensor noise camera identification: countering counter-forensics. In *Proceedings of the SPIE, Media Forensics and Security II*, volume 7541, 2010.

[78] M. Goljan, J. Fridrich, and M. Chen. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security*, 6(1):227–236, March 2011.

[79] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[80] X. Guo, X. Liu, E. Zhu, and J. Yin. Deep clustering with convolutional autoencoders. In D. Liu, S. Xie, Y. Li, D. Zhao, and El-Sayed M. El-Alfy, editors, *Neural Information Processing*, pages 373–382, Cham, 2017. Springer International Publishing.

[81] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, (NIPS)*, pages 1025–1035, 2017.

[82] H. Han and A. K. Jain. Age, gender and race estimation from unconstrained face images. 2014.

[83] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[84] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019.

[85] K. Hernandez-Diaz, F. Alonso-Fernandez, and J. Bigun. Periocular recognition using CNN features off-the-shelf. In *Proceedings of the 17th International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5, September 2018.

[86] E. Hoffar and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[87] V. Holub, J. Fridrich, and Tomás Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014:1–13, 2014.

[88] G. B. Huang, M. R., T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[89] B. Jähne and H. Haußecker, editors. *Computer Vision and Applications: A Guide for Students and Practitioners*. Academic Press, Inc., Orlando, FL, USA, 2000.

[90] A. K. Jain, A. Ross, and K. Nandakumar. Introduction to biometrics. *Springer*.

[91] A. K. Jain, A. Ross, and K. Nandakumar. Introduction to biometrics. *Springer*, 2011.

[92] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, Jan 2004.

[93] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., USA, 1988.

[94] R. Jillela and A. Ross. Ocular and periocular biometrics. In *Wiley Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Inc., 2016.

[95] R. Jillela, A. Ross, V. N. Boddeti, B. V. K. Vijaya Kumar, X. Hu, R. Plemmons, and P. Pauca. Iris segmentation for challenging periocular images. In K. W. Bowyer and Mark J. Burge, editors, *Handbook of Iris Recognition*, pages 281–308. Springer, London, 2016.

[96] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *Trans. Img. Proc.*, 6(7):965–976, July 1997.

[97] F. Juefei-Xu and M. Savvides. Subspace-based discrete transform encoded local binary patterns representations for robust periocular matching on NIST's Face Recognition Grand Challenge. *IEEE Transactions on Image Processing*, 23(8):3490–3505, Aug 2014.

[98] N. Kalka, N. Bartlow, B. Cukic, and A. Ross. A preliminary study on identifying sensors from iris images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 50–56, June 2015.

[99] X. Kang, Y. Li, Z. Qu, and J. Huang. Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 7(2):393–402, April 2012.

[100] C. Kauba, L. Debiasi, and A. Uhl. Identifying the Origin of Iris Images Based on Fusion of Local Image Descriptors and PRNU Based Techniques. In *3rd International Joint Conference on Biometrics (IJCB)*, October 2017.

[101] Yan Ke, Rahul Sukthankar, and Larry Huston. An efficient parts-based near-duplicate and sub-image retrieval system. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 869–876, 2004.

[102] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, (ICLR),*, 2017.

[103] J. Komulainen, Z. Boulkenafet, and Z. Akhtar. Review of face presentation attack detection competitions. In Sébastien Marcel, Mark S. Nixon, Julian Fierrez, and Nicholas Evans, editors, *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection*, pages 291–317. Springer International Publishing, 2019.

[104] Guoqi L. and Changyun W. Legendre polynomials in signal reconstruction and compression. In *5th IEEE Conference on Industrial Electronics and Applications*, pages 1636–1640, June 2010.

[105] B. Levy. Review of "Digital image forensics: There is more to a picture than meets the eye" by Husrev Taha Sencar and Nasir Memon (Editors). volume 4, page 17. 2013.

[106] C. T. Li. Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287, June 2010.

[107] C. T. Li, C. Y. Chang, and Y. Li. On the repudiability of device identification and image integrity verification using sensor pattern noise. pages 19–25, 2010.

[108] X. Lin and C. T. Li. Enhancing sensor pattern noise via filtering distortion removal. *IEEE Signal Processing Letters*, 23(3):381–385, March 2016.

[109] X. Lin and C. T. Li. Preprocessing reference sensor pattern noise via spectrum equalization. *IEEE Transactions on Information Forensics and Security*, 11(1):126–140, Jan 2016.

[110] S. P. Lloyd. Least squares quantization in pcm. Technical Report RR-5497, Bell Lab, 1957.

[111] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Vol.2*, IJCAI, pages 674–679. Morgan Kaufmann Publishers Inc., 1981.

[112] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

[113] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

[114] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2000: fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, March 2002.

[115] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. A deep learning approach for iris sensor model identification. *Pattern Recognition Letters*, 2017.

[116] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. A deep learning approach for iris sensor model identification. *Pattern Recognition Letters*, 113:46 – 53, 2018.

[117] M. D. Marsico, M. Nappi, F. Narducci, and H. Proença. Insights into the results of MICHE I - mobile iris challenge evaluation. *Pattern Recognition*, 74:286 – 304, 2018.

[118] J. R. Matey. Iris device. In S. Z. Li and A. K. Jain, editors, *Encyclopedia of Biometrics*, pages 774–778. Springer, 2009.

[119] A. Melloni, P. Bestagini, S. Milani, M. Tagliasacchi, A. Rocha, and S. Tubaro. Image phylogeny through dissimilarity metrics fusion. In *Fifth European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, Dec 2014.

[120] R. Mercuri. Courtroom considerations in digital image forensics. In H. T. Sencar and N. Memon, editors, *Digital Image Forensics: There is More to a Picture than Meets the Eye*, pages 313–325. Springer New York, New York, NY, 2013.

[121] S. Milani, M. Fontana, P. Bestagini, and S. Tubaro. Phylogenetic analysis of near-duplicate images using processing age metrics. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2054–2058, March 2016.

[122] V. Mirjalili, S. Raschka, and A. Ross. Gender privacy: An ensemble of semi adversarial networks for confounding arbitrary gender classifiers. In *9th IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10, Oct 2018.

[123] V. Mirjalili and A. Ross. Soft biometric privacy: Retaining biometric utility of face images while perturbing gender. In *Proc. of IEEE International Joint Conference on Biometrics (IJCB)*, pages 564–573, Oct 2017.

[124] D. Moreira, A. Bharati, J. Brogan, A. Pinto, M. Parowski, K. W. Bowyer, P. J. Flynn, A. Rocha, and W.. Scheirer. Image provenance analysis at scale. *IEEE Transactions on Image Processing (T-IP)*, 27(12), 2018.

[125] S. Nagaraja, P. Schaffer, and D. Aouada. Who clicks there!: Anonymising the photographer in a camera saturated society. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, pages 13–22, New York, NY, USA, 2011.

[126] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, page 849–856, 2001.

[127] L. Nie, A. Kumar, and S. Zhan. Periocular recognition using unsupervised convolutional rbm feature learning. In *22nd International Conference on Pattern Recognition*, pages 399–404, Aug 2014.

[128] M. A. Oikawa, Z. Dias, A. de Rezende Rocha, and S. Goldenstein. Manifold learning and spectral clustering for image phylogeny forests. *IEEE Transactions on Information Forensics and Security*, 11(1):5–18, Jan 2016.

[129] A. Oliveira, P. Ferrara, A. De Rosa, A. Piva, M. Barni, S. Goldenstein, Z. Dias, and A. Rocha. Multiple parenting identification in image phylogeny. In *IEEE International Conference on Image Processing (ICIP)*, pages 5347–5351, Oct 2014.

[130] S. Omachi and M. Omachi. Fast template matching with polynomials. *IEEE Transactions on Image Processing*, 16(8):2139–2149, Aug 2007.

[131] C. N. Padole and H. Proenca. Compenstaing for pose and illumination in unconstrained periocular biometrics. *International Journal of Biometrics*, 5(3):336–359, 2013.

[132] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[133] N. Le Philippe, W. Puech, and C. Fiorio. Phylogeny of jpeg images by ancestor estimation using missing markers on image pairs. In *Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, Dec 2016.

[134] A. C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, Oct 2005.

[135] Salil Prabhakar, Alexander Ivanisov, and A.K. Jain. Biometric recognition: Sensor characteristics and image quality. *Instrumentation & Measurement Magazine, IEEE*, 14:10 – 16, 07 2011.

[136] G. W. Quinn, J. Matey, E. Tabassi, and P. Grother. IREX V: Guidance for iris image collection. In *NIST Intragency Report 8013*, 2014.

[137] J. Redi, W. Taktak, and J.L. Dugelay. Digital image forensics: A booklet for beginners. *Multimedia Tools and Applications*, 51(1):133–162, 2011.

[138] J. A. Redi, W. Taktak, and J. L. Dugelay. Digital image forensics: A booklet for beginners. *Multimedia Tools Appl.*, 51(1):133–162, January 2011.

[139] A. Ross, S. Banerjee, C. Chen, A. Chowdhury, V. Mirjalili, R. Sharma, T. Swearingen, and S. Yadav. Some research problems in biometrics: The future beckons. In *12th IAPR International Conference on Biometrics (ICB)*, Crete, Greece, June 2019.

[140] C. Sammut. Density estimation. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 348–349. Springer US, Boston, MA, 2017.

[141] M. Ester H. Kriegel J. Sander and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[142] Gerald Schaefer and Michal Stich. UCID: an uncompressed color image database. In Minerva M. Yeung, Rainer W. Lienhart, and Chung-Sheng Li, editors, *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472 – 480. International Society for Optics and Photonics, SPIE, 2003.

[143] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015.

[144] S. A. C. Schuckers, N. A. Schmid, A. Abhyankar, V. Dorairaj, C. K. Boyce, and L. A. Hornak. On techniques for angle compensation in nonideal iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1176–1190, Oct 2007.

[145] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 41–48. MIT Press, 2003.

[146] E. Shutova, S. Teufel, and A. Korhonen. Statistical Metaphor Processing. *Comput. Linguist.*, 39(2):301–353, June 2013.

[147] R. Singh, M. Vatsa, and A. Noore. Improving verification accuracy by synthesis of locally enhanced biometric images and deformable model. *Signal Processing*, 87(11):2746 – 2764, 2007.

[148] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

[149] N. A. Spaun. Forensic biometrics from images and video at the federal bureau of investigation. In *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, pages 1–3, Sep. 2007.

[150] G. Stockman and L. G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[151] V. Štruc and N. Pavesic. Photometric normalization techniques for illumination invariance. In *Advances in Face Image Analysis: Techniques and Technologies*, pages 279–300. IGI-Global, 01 2011.

[152] C. W. Tan and A. Kumar. Towards online iris and periocular recognition under relaxed imaging constraints. *IEEE Transactions on Image Processing*, 22(10):3751–3765, Oct 2013.

[153] L. W. Tu. Bump functions and partitions of unity. In *An Introduction to Manifolds*, pages 127–134. Springer New York, New York, NY, 2008.

[154] A. Uhl and Y. Höller. Iris-sensor authentication using camera PRNU fingerprints. In *5th IAPR International Conference on Biometrics (ICB)*, pages 230–237, March 2012.

[155] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli. User authentication via PRNU-based physical unclonable functions. *IEEE Transactions on Information Forensics and Security*, 12(8):1941–1956, 2017.

[156] L. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2431–2556, November 2008.

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

[158] M. Vollmer and K. P. Möllmann. Infrared thermal imaging: Fundamentals, research and applications. *New York: Wiley*, 2010.

[159] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, Feb 1992.

[160] Haitao Wang, S. Z. Li, and Yangsheng Wang. Face recognition under varying lighting conditions using self quotient image. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 819–824, May 2004.

[161] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5005–5013, 2016.

[162] C. Ye, R. C. Wilson, C. H. Comin, L. da F. Costa, and E. R. Hancock. Entropy and heterogeneity measures for directed graphs. In E. Hancock and M. Pelillo, editors, *Similarity-Based Pattern Recognition*, pages 219–234, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[163] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[164] K. Zuiderveld. Contrast Limited Adaptive Histogram Equalization. In Paul S. Heckbert, editor, *Graphics Gems IV*, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.